HCLSoftware

HCL Z Data Tools
User's Guide and Reference for Db2 Data
Version 17.1.0



Note

Before using this information and the product it supports, be sure to read the general information under Notices on page mcxci .

Edition Notice

This edition (published October 2026) applies to Version 17 Release 1 Modification Level 0 of HCL Z Data Tools (program number HCL19OP1220), and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this document..... xiii
Who should use this document.....xiv
National characters..... xiv
Prerequisite and related information..... xv
How to read syntax diagrams..... xv
Summary of changes.....xix
Part I. Working with ZDT/Db2..... 20
Chapter 1. ZDT/Db2 overview..... 21
What you can do with ZDT/Db2.....21
Templates..... 21
What is a template?..... 21
Where you can use templates..... 22
Chapter 2. System requirements for Z Data Tools for Db2®..... 24
Chapter 3. Getting started with ZDT/Db2.....25
Starting and exiting ZDT/Db2..... 25
TSO region size..... 25
Starting ZDT/Db2..... 26
Checking your ZDT/Db2 version..... 27
Checking which COBOL compiler you're using.....29
Exiting from ZDT/Db2..... 29
Checking which PL/I compiler you are using 29
Selecting the Db2® subsystem when more than one is available.....30
Selecting the Db2® subsystem by using the Db2 SSID field.....30
Selecting the Db2® subsystem by entering the SSID command.....32
Using the Db2® Subsystem Selection menu.....33
Using the ZDT/Db2 interface..... 35
Navigating in ZDT/Db2..... 35
ZDT/Db2 panel features..... 38
Screen size.....40
Scrollable input and display fields for long names.....40
Selecting options on ZDT/Db2 panels..... 45
Specifying a Db2® object name.....46
Specifying the SQLID..... 52
Specifying a data set and a member name..... 52
Canceling long-running Db2® queries..... 56
Getting help..... 57
Accessing Tutorial Help.....57
Displaying field-level help information..... 60
Displaying the expanded form of error messages.....60
Setting default processing options.....62
System Options (option 0).....64
Print settings (option 0.1).....64
ZDT/Db2 system options (option 0.2)..... 65
Job card specifications (option 0.3)..... 65
Compiler language selection (option 0.4).....65
COBOL compiler specifications (option 0.5)..... 65
HLASM compiler specifications (option 0.6).....66
PL/I compiler specifications (option 0.7)..... 66
Temporary Data Set Allocations (option 0.8)..... 66
Output Data Set Allocations (option 0.9)..... 67
Trace options (option 0.10)..... 67
ISPF settings (option 0.11).....67
View and Edit options (options 1 and 2).....67
Utilities (option 3).....68
Print utility options (option 3.1)..... 68
Copy utility options (option 3.3)..... 68
Object List Options (option 3.4)..... 68
Export utility options (option 3.7).....69
Db2® LOAD utility options (option 3.L).....69
Db2® Utility LISTDEF options (option 3.UL)..... 69
Db2® Utility OPTIONS options (option 3.UO)..... 70
Db2® Utility TEMPLATE options (option 3.UT)..... 70
Db2® Unload utility options (option 3.UU)..... 70
Displaying or editing DBCS characters..... 70
SAF-rule controlled auditing.....71
Determining when SAF-rule controlled auditing is in effect..... 71
SAF rules for auditing..... 72
SAF-rule controlled auditing not in effect..... 72
SAF-rule controlled auditing in effect, no auditing.....72
SAF-rule controlled auditing in effect, auditing possible..... 72
Chapter 4. Working with templates..... 74
Associating a template with a Db2® object..... 74
Information stored in a template.....75
Printing a template.....77
Using templates with non-Db2 data..... 77
How ZDT/Db2 compiles a copybook into a template..... 80
Support for COBOL data description entries..... 81
Support for HLASM data description entries.....82
Support for PL/I data description entries..... 82
Support for variable-length arrays.....83
Editing a template.....84
Adding a description to a template.....85
Selecting rows.....86

Row selection by column....86
Row selection criteria in freeform style....92
Scrolling the Row Selection Criteria field....94
Expanding the Row Selection Criteria field....95
Using the SQL Edit session....95
Manipulating columns....98
Selecting columns....99
Holding columns on the display when scrolling....100
Changing the sequence of displayed or printed data....101
Changing column attributes....104
ROWID columns....107
Specifying scrambling options....108
Selecting a column for specifying scrambling options....108
Mapping data....112
Where you can map data....113
Letting ZDT/Db2 do the mapping....113
Specifying your own mapping....114
Column mapping rules....118
Creating a copybook from a Z Data Tools/Db2 Template....121
Chapter 5. Viewing and changing Db2® data....122
ZDT/Db2 editor sessions....123
Starting a Z Data Tools/Db2 editor session....123
Specifying the editor session mode: "normal mode" or "large mode"....124
Specifying the start position for your ZDT/Db2 editor session....126
Setting options for the current ZDT/Db2 editor session....126
Validating and saving changes with a Z Data Tools/Db2 editor session....127
Ending a Z Data Tools/Db2 editor session....133
Starting another ZDT/Db2 editor session....135
Specifying the Db2® object you want to view....136
Restricting the data that ZDT/Db2 loads from Db2®....138
Storage considerations....138
Viewing large tables or views....139
Viewing data in a Z Data Tools/Db2 editor session....141
Count of rows fetched in a Z Data Tools/Db2 editor session....141
Locating a row....142
Locating a column....142
Finding a string....146
Finding data within specific columns....147
Finding a string within specific portions of the data....149
Finding data within excluded or non-excluded rows....149
Finding data within a label range....150
Searching numeric columns....152
Changing the view of displayed data....152
Handling special data....167
DATE and TIME columns....168
Large numeric fields....168
Working with large tables....169
Null values and VARCHAR processing....178
Unicode data....179
Changing data....180
Installation settings that affect CAPS....180
Changing data by overtyping....181
Changing data with the CHANGE command....182
Changing data using prefix commands....186
Converting data to uppercase....189
"Read-only" views, non-updateable columns....190
Deleting rows....191
Deleting rows within excluded or non-excluded rows....192
Deleting rows within a label range....192
Editing related tables....192
Listing related tables....193
Self-referencing constraints....195
Referential integrity errors....195
Starting an Edit session of a related table....195
Differences when you edit a related table....196
Differences when saving or canceling while editing a related table....196
Navigating between Edit sessions....197
Deleting a primary key....197
Inserting a new foreign key....198
Updating a primary key....199
Adding a primary key....200
Issuing Db2® commands from within a Z Data Tools/Db2 session....200
Issuing Db2® commands from the Enter and Execute Db2® Commands panel....200
Issuing Db2® commands from the command line....201
Handling errors during an Edit session....202
Determining why an error occurred....203
"Duplicate Row" errors....203
"No Primary Key" errors....203
"Delete Restrict Rule" errors....204
"Check Constraint" errors....204
"Row Not Found" errors....205
Deadlocks or timeouts....205

Example of correcting an error during an Edit session....205
Chapter 6. Creating and dropping Db2® objects....208
Working with the Create and Drop panels....209
SQL Keywords....210
Creating a database....223
Creating a table space....224
Table Space Type (Create Table Space)....224
Table Space Allocation (Create Table Space)....225
Data Storage (Create Table Space)....225
Locking (Create Table Space)....225
Data Sharing (Create Table Space)....225
Partitions (Create Table Space)....226
Creating a table....226
Creating a table using a model....227
Creating a table without using a model....230
Table Creation options....230
1. Columns (Create Table)....231
2. Nulls and default values (Create Table)....231
3. Options (Create Table)....232
4. Unique constraints (Create Table)....232
5. Referential constraints (Create Table)....232
6. Check constraints (Create Table)....234
7. Procedure exits (Create Table)....235
8. Generate values (Create Table)....235
Creating a view....235
Creating an alias....236
Creating an index....236
Index type (Create Index)....237
Index Column Selection (Create Index)....237
Allocation (Create Index)....237
Index (Create Index)....238
Index Partitions (Create Index)....238
Partition Values Option (Create Index)....238
Creating a synonym....238
Creating a distinct type....239
Creating a function....239
Parameters (Create Function)....239
Parameter Types (Create Function)....240
Returned Data Type (Create Function)....240
Option List 1/2 (Create Function)....240
Option List 2/2 (Create Function)....240
Creating a procedure....240
Procedure Parameters (Create Procedure)....241
Parameter Types (Create Procedure)....241
Option List 1/2 (Create Procedure)....241
Option List 2/2 (Create Procedure)....241
Creating a trigger....241
Details (Create Trigger)....242
Search Conditions (Create Trigger)....242
SQL Statement (Create Trigger)....242
Creating an auxiliary table....243
Dropping Db2® objects....243
Chapter 7. Populating a Db2® table with data....244
Editing the template....245
Chapter 8. Copying data....246
Copying data from one Db2® object to another....246
Specifying details for the "From" table....247
Specifying details for the "To" table....249
Copying data from a VSAM or QSAM file....251
When (and when not) to use the Import utility....252
Specifying details for the "From" data set....253
Specifying details for the "To" table....255
Using a REXX procedure with the Import utility....257
General information about REXX....258
Enhancing the Import utility using a REXX procedure....268
Copying data to a VSAM or QSAM file....279
Specifying details for the "From" table....281
Specifying details for the "To" data set....284
Copying data from a Z Data Tools/Db2 editor session, or from an SQL statement....288
Scrambling data....292
When can you scramble data?....292
Scrambling rules....292
How do I...?....292
How do I copy data from one Db2® table to another on the same Db2® system?....293
How do I copy data from one Db2® table to another on a different (connectable) Db2® system?....293
How do I copy data from one Db2® table to another on a different (non-connectable) Db2® system?....293
How do I copy and reorder selected data from a Db2® table to a sequential file?....294
How do I define a copybook that describes data exported to a sequential file?....295
How do I export data using a copybook that describes a different data format to that used by the ZDT/Db2 Export Utility?....301
Chapter 9. Working with lists of Db2® objects....303
Displaying or printing a list of objects....304
Working with object list panels....307
Limiting the data displayed....309
Locating a row or column in a list of Db2® objects....314
Finding a string in a list of Db2® objects....314
Using the line command area (Cmd)....314

Sorting the displayed data....323
Reverse engineering....324
Performance tips....326
Chapter 10. Printing....328
Printing the contents of a Db2® table....328
Formatting the print output....329
Editing the template....331
Running Print in batch....331
Printing rows from a Db2® table....331
Looking at the print output from your ZDT/Db2 session....333
Printing an audit trail report....335
Audit Trail Report....336
Chapter 11. Managing Db2® privileges....339
Using the Manage Db2® Privileges panel....340
Displaying privileges....342
Granting privileges....342
Revoking privileges....344
Chapter 12. Generating batch JCL for Db2® utility jobs....346
Setting options for Db2® utilities....348
Generating a LISTDEF statement in the Db2® utility job....350
Generating a Db2® utility job....350
Working with LISTDEF utility panels....353
COPY utility....354
Default values....354
COPY (table spaces)....355
COPY (index spaces)....356
LOAD utility....357
Input data....358
Loading data using LOAD utility control statements....359
Loading data described by a copybook or template....361
REBUILD utility....365
REBUILD utility (indexes)....366
REBUILD utility (indexes for table spaces)....367
RECOVER utility....368
RECOVER (table spaces)....368
RECOVER (index spaces)....370
REORG utility....371
REORG (table spaces)....372
REORG (indexes)....374
RUNSTATS utility....376
RUNSTATS (table spaces)....377
RUNSTATS (indexes)....378
UNLOAD utility....380
UNLOAD (tables)....381
UNLOAD (table spaces)....386
Chapter 13. Working with SQL statements....390
Should you use basic or advanced SQL prototyping?....390
Using basic SQL prototyping....390
Specifying a predicate....394
Example of using basic SELECT prototyping....395
Using advanced SQL prototyping....399
Tabbing and color-coding....401
Using the advanced SELECT prototyping panels to build a SELECT statement....402
Changing the SQL statement....407
Handling repeatable items in the syntax....416
Executing a SELECT statement after you have built it using prototyping....424
Entering, executing, and explaining SQL statements....424
Entering SQL statements....425
Executing SQL statements....426
Explaining SQL....427
Editing and executing SQL statements from a data set....429
Managing tables used by SQL explain....430
Exiting SQL prototyping....430
Chapter 14. Using ZDT/Db2 functions in batch jobs....432
JCL syntax for Z Data Tools/Db2 batch jobs....432
Using Z Data Tools/Db2 edit models....433
Edit model example: JCL that calls the DBP function....434
Chapter 15. Interfacing with Db2® Interactive....438
Part II. ZDT/Db2 reference....439
Chapter 16. Db2® data types....440
Large objects (LOBs)....444
Chapter 17. ZDT/Db2 panels and fields....445
Scrollable fields on ZDT/Db2 panels....445
Action bar pull-down menu....445
Additional LISTDEF Statements panel....449
Additional TEMPLATE Statements panel....450
Advanced SELECT Prototyping panel....451
Alter Database panel....453
Alter External Scalar Function panel....454
Alter Index panel....455
Alter Stored Procedure panel....456
Alter Table panel....457
Alter Table - ADD CHECK Constraint panel....458
Alter Table - ADD Column panel....459
Alter Table - ADD FOREIGN KEY Constraint panel....460
Alter Table - ADD PRIMARY KEY panel....461
Alter Table - AUDIT panel....462

Alter Table - DATA CAPTURE panel..... 463
Alter Table - DROP CHECK Constraint panel..... 464
Alter Table - DROP FOREIGN KEY Constraint panel..... 465
Alter Table - DROP PRIMARY KEY panel..... 466
Alter Table - RESTRICT ON DROP panel..... 467
Alter Table - VALIDPROC panel..... 468
Alter Table Space panel..... 469
Application Packages panel..... 469
Application Plans panel..... 473
Basic SELECT Prototyping panel..... 476
Bind Application Plan panel..... 481
Bind Package panel..... 482
Children of Table panel..... 482
Collections panel..... 482
Column Attributes panel (alphanumeric)..... 484
Column Attributes panel (DATE)..... 489
Column Attributes panel (numeric)..... 490
Column Attributes panel (TIME)..... 494
Column Attributes panel (TIMESTAMP)..... 495
Column Distribution panel..... 497
Column Part Statistics panel..... 497
Column Selection/Edit panel..... 497
Columns panel..... 503
Columns in Index panel..... 505
Columns in Table panel..... 505
Comment panels..... 506
Compiler Language Selection panel..... 506
Constraint Columns for panel..... 509
Copy Options panel..... 509
Copy Utility ("From") panel..... 513
COPY Utility (Index Spaces) panel..... 517
COPY Utility (Index Spaces) with LISTDEF panel..... 519
Copy Utility ("To") panel..... 521
COPY Utility (Table Spaces) panel..... 524
COPY Utility (Table Spaces) with LISTDEF panel..... 526
Create Alias panel..... 527
Create Auxiliary Table panel..... 528
Create Database panel..... 529
Create Distinct Type panel..... 531
Create Function: Option List (1/2) panel..... 533
Create Function: Option List (2/2) panel..... 534
Create Function panel..... 536
Create Function: Parameter Types panel..... 537
Create Function: Parameters panel..... 539
Create Function: Returned Data Type panel..... 540
Create Index panel..... 542
Create Index: Allocation panel..... 545
Create Index: Column Selection panel..... 546
Create Index: Options panel..... 547
Create Index: Partitions panel..... 549
Create Index: Partition Values panel..... 550
Create Index: Type panel..... 552
Create Procedure panel..... 553
Create Procedure: Option List (1/2) panel..... 554
Create Procedure: Option List (2/2) panel..... 556
Create Procedure: Parameter Types panel..... 557
Create Procedure: Parameters panel..... 558
Create Synonym panel..... 560
Create Table panel..... 561
Create Table Space panel..... 564
Create Table Space: Allocation panel..... 566
Create Table Space: Data Sharing Options panel..... 567
Create Table Space: Data Storage Options panel..... 568
Create Table Space: Define Partitions panel..... 570
Create Table Space: Locking Options panel..... 572
Create Table Space: Type panel..... 573
Create Table: Column Check Constraints panel..... 575
Create Table: Column Referential Constraints panel..... 576
Create Table: Columns panel..... 578
Create Table: Generate Values panel..... 580
Create Table: Model Load panel..... 582
Create Table: Nulls and Default Values panel..... 584
Create Table: Options panel..... 587
Create Table: Procedure Exits panel..... 589
Create Table: Table Check Constraints panel..... 590
Create Table: Table Referential Constraint panel..... 592
Create Table: Table Referential Constraints panel..... 593
Create Table: Unique Constraints panel..... 595
Create Trigger panel..... 596
Create Trigger: Details panel..... 598
Create Trigger: Search Condition panel..... 600
Create Trigger: SQL statement panel..... 600
Create View panel..... 601
Create View Source Statements panel..... 603
Data Create Utility panel..... 603
Database Request Modules panel..... 607
Database Structure panel..... 609
Databases panel..... 609
Db2® Browse panel..... 612
Db2® Edit panel..... 616
Db2® Location Selection panel..... 621
Db2® Object Functions panel..... 622
Db2® Save Error Action panel..... 624
Db2® Subsystem Selection panel..... 624

Db2® Template Description panel..... 626
Db2® Utilities panel.....627
Db2® View panel..... 631
Db2I Primary Option Menu panel..... 635
Details panels.....636
Display Row panel..... 637
Distinct Types panel..... 637
Drop panels..... 640
Drop Confirmation panel..... 640
Edit/Execute SQL (Data Set) panel.....641
Editor Options (1 of 8) panel..... 644
Editor Options (2 of 8) panel..... 649
Editor Options (3 of 8) panel..... 655
Editor Options (4 of 8) panel..... 662
Editor Options (5 of 8) panel..... 667
Editor Options (6 of 8) panel..... 672
Editor Options (7 of 8) panel..... 675
Editor Options (8 of 8) panel..... 680
Enter and Execute Db2® Commands panel.....683
Enter, Execute and Explain SQL Statements panel.... 684
EXPLAIN Interpretation panel..... 686
Explain Utilities panel..... 686
Export Options (1 of 3) panel..... 689
Export Options (2 of 3) panel..... 692
Export Options (3 of 3) panel..... 696
Export Utility ("From") panel.....698
Export Utility ("To") panel..... 701
Field Selection/Edit panel..... 704
ZDT/Db2 Release News panel.....708
Foreign Keys for Table panel..... 708
Free Application Plan panel..... 708
Free Package panel..... 708
From Column Mapping panel..... 709
Functions panel.....709
Generate SQL From Db2® Catalog panel.....712
Grant privileges panels.....715
Import Options panel.....716
Import Utility ("From") panel.....719
Import Utility ("To") panel..... 720
Index Parts panel.....723
Indexes panel.....723
Indexes and Columns for panel.....726
ISPF Primary Option panel.....727
ISPF Settings panel..... 727
Label Column panel.....728
Label Table panel.....728
LISTCAT Output Display panel.....729
LISTDEF Options panel.....729
LOAD from panel..... 731
LOAD Utility panel..... 733
LOAD Utility panel using Db2® templates.....734
LOAD Utility - Using Templates panel.....735
LOAD Utility Options panel.....737
Manage Db2® Privileges panel..... 741
Object List Line Commands panel..... 742
Object List Options (1 of 2) panel..... 743
Object List Options (2 of 2) panel..... 746
Object List Utility panel..... 748
OPTIONS Options panel..... 750
Package Dependencies panel.....753
Package List panel..... 753
Parents of Table panel..... 753
Partitioned Table Statistics panel.....753
Plan Dependencies panel.....753
Plan Enabled/Disabled Connections panel..... 754
Plan Table Rows panel.....754
Primary Key for Table panel.....760
Primary Option Menu panel..... 760
Print Audit Trail panel.....761
Print Utility Options panel..... 763
Print Utility panel.....764
Privilege details panels..... 768
Privileges panels.....768
Rebind Application Plan panel.....770
Rebind Package panel.....772
REBUILD (Indexes for Table Spaces) panel..... 773
REBUILD (Indexes for Table Spaces) with LISTDEF panel..... 775
REBUILD Utility (Indexes) panel.....777
REBUILD (Indexes) with LISTDEF panel..... 779
Record Type Selection panel..... 780
RECOVER Utility (Index Spaces) panel..... 780
RECOVER Utility (Index Spaces) with LISTDEF panel..... 782
RECOVER Utility (Table Spaces) panel..... 783
RECOVER Utility (Table Spaces) with LISTDEF panel..... 785
Recovery Information panel..... 787
Related tables panel.....787
Remote Db2® Location Selection panel..... 791
REORG Utility (Indexes) panel.....792
REORG Utility (Indexes) with LISTDEF panel.....793
REORG Utility (Table Spaces) panel..... 795
REORG Utility (Table Spaces) with LISTDEF panel.....797
Revoke privileges panels.....798

Routine Parameters panel....799
Row Selection Criteria panel....799
RUNSTATS Utility (Indexes) panel....805
RUNSTATS Utility (Indexes) with LISTDEF panel....807
RUNSTATS Utility (Table Spaces) panel....808
RUNSTATS Utility (Table Spaces) with LISTDEF panel....810
Schemas panel....812
Scramble Exit Specification panel....814
Select Statement Browse panel....815
Select Statement Edit panel....815
Sequences panel....816
Set Batch Job Card Information panel....817
Set COBOL Processing Options panel....818
Set HLASM Processing Options panel....821
Set Optimizer Hint panel....823
Set Output Data Set Allocation Options panel....824
Set PL/I Processing Options panel....825
Set Print Processing Options panel....827
Set Processing Options panel....832
Set System Processing Options panel....835
Set Temporary Data Set Allocation Options panel....837
Set Trace options panel....839
Set Utility Options panel....840
Sort Fields panel....841
SQL Prototyping, Execution and Analysis panel....843
Statement Table Rows panel....845
Storage Group(s) panel....848
Storage Group Usage Privileges panel....851
Stored Procedures panel....852
Synonym Dependencies panel....855
Synonyms panel....855
Systems Options (1 of 4) panel....857
Systems Options (2 of 4) panel....859
Systems Options (3 of 4) panel....862
Systems Options (4 of 4) panel....864
Table Browse panel....866
Table Check Constraint Dependencies panel....868
Table Check Constraint Details panel....868
Table Check Constraints panel....868
Table Edit panel....868
Table Edit (related) panel....877
Table Selection panel....878
Table Space Parts panel....879
Table Spaces panel....879
Table View panel....882
Table/View/Alias Selection panel....890
Tables, Views and Aliases panel....891
TEMPLATE Options (1 of 2) panel....897
TEMPLATE Options (2 of 2) panel....898
Template Mapping panel....900
Template Save panel....902
Triggers panel....903
UNLOAD Options panel....905
UNLOAD Utility (Table Spaces) panel....908
UNLOAD Utility (Table Spaces) with LISTDEF panel....909
UNLOAD Utility (Tables) "From" panel....910
UNLOAD Utility (Tables) "To" panel....911
UNLOAD Utility (Tables) with LISTDEF panel....914
Utility Functions panel....915
Value List Edit panel....917
Volumes panel....919
Chapter 18. ZDT/Db2 commands....922
ABOUT primary command....922
ALL primary command....922
BACKWARD primary command....923
BOTTOM primary command....924
BROWSE primary command....925
CANCEL primary command....926
CAPS primary command....926
CASE primary command....927
CHANGE primary command....928
COL primary command....934
CHARPOS primary command....935
DELCOL primary command....935
DELETE primary command....936
DESCRIBE primary command....938
DESELECT primary command....938
DOWN primary command....939
EDIT primary command....940
EDITCOL primary command....941
END primary command....942
EXCLUDE primary command....943
EXECUTE primary command....948
EXPAND primary command....949
EXPLAIN primary command....950
EXPORT primary command....950
FILE primary command....953
FIND primary command....953
FMAP primary command....961
FORMAT primary command....962
FORWARD primary command....963
FREE primary command....964
FS primary command....965
FT primary command....966

HEX primary command....966
HIDE primary command....967
HINT primary command....968
HOLD primary command....968
INDEXINF primary command....970
INSRPT primary command....971
LEFT primary command....971
LHEX primary command....973
LOBBRWS primary command....974
LOBEDIT primary command....975
LOBLEN primary command....976
LOBVIEW primary command....977
LOCATE primary command....978
LZERO primary command....982
NEW, NEWS primary command....983
NEXT primary command....983
NEXTRPT primary command....984
ORDER primary command....984
PB primary command....986
PLAN primary command....986
PREFIX primary command....987
PREVIOUS primary command....988
PREVRPT primary command....988
PRINT primary command....989
PROTCOL primary command....990
PURGE primary command....990
QUIT primary command....991
RCHANGE primary command....991
RD primary command....991
REDIT primary command....992
REFRESH primary command....993
REFS primary command....994
RESET primary command....994
RFIND primary command....995
RIGHT primary command....996
RP primary command....997
RUNTEMP primary command....998
SAVE primary command....999
SAVEAS primary command....999
SHADOW primary command....1000
SHOW primary command....1001
SHOW primary command....1001
SHOWCOB primary command....1003
SHOWPLI primary command....1003
SORT primary command....1003
SQL primary command....1005
SQLID primary command....1006
TEDIT primary command....1006
TOP primary command....1007
TP primary command....1007
TYPE primary command....1008
UNDO primary command....1008
UP primary command....1009
VER primary command....1010
VIEW primary command....1010
WIDTH primary command....1011
XMLBRWS primary command....1013
XMLEDIT primary command....1014
XMLLEN primary command....1015
XMLVIEW primary command....1015
ZOOM primary command....1016
Chapter 19. ZDT/Db2 functions....1018
How to use this reference chapter....1018
General tips about performance when you use Z Data Tools functions....1018
ZDT/Db2 batch functions....1019
BATSQL (Batch SQL) batch command....1019
DBC (Copy) batch command....1027
DBI (Import) batch command....1034
DBP (Print) batch command....1040
DBX (Export) batch command....1043
D2G (Data Generate) batch command....1058
D2TP (Template create/update) batch command....1060
Specifying a Db2® object name....1063
Specifying the template for a Db2® object....1064
Specifying the template/copybook for a data set....1064
Appendix A. Messages....1066
ZDT/Db2 messages....1066
Notices....mcxci
Programming interface information....mcxciii
Index....1194

User's Guide and Reference for Db2 Data

This publication provides guidance and reference information for users of Z Data Tools Db2® component (ZDT/Db2).

About this document

This document provides guidance and reference information for users of Z Data Tools Db2® component (ZDT/Db2).

This document consists of three parts:

Part 1. Working with ZDT/Db2

Introduces key concepts and provides step-by-step instructions on how to perform tasks using the ZDT/Db2 panels under ISPF. The information is presented in a task-based format, with panel and command definitions provided when they are relevant to the current task. For a complete description of a particular panel or command, refer to the appropriate chapter in **Part 2 ZDT/Db2 reference**.

Part 1 can be used both as a learning tool for new users and as a way to quickly find specific task instructions.

Part 2. ZDT/Db2 reference

Provides a lookup reference for ZDT/Db2 panels and commands. The information is presented in two chapters:

- **ZDT/Db2 panels and fields**: Lists the ZDT/Db2 panels, alphabetically by panel title. For each listed panel, cross-references are provided to one or more *"parent"* panels (the panels from which you can reach the panel in question) and also, if applicable, cross-references to any *"child"* panels (the panels you can reach from this panel). For several of the panels, there is a definition for each field in the panel including, where applicable, the value ranges that are valid for each field.
- **ZDT/Db2 commands**: Lists the primary commands you can use with ZDT/Db2, including a description of their syntax and usage.

Both new and experienced users of ZDT/Db2 can use Part 2 to quickly look up such things as the exact syntax of a command or the acceptable values for a field in a panel.

Part 3. Appendix

"ZDT/Db2 messages", provides the full text explanation of all ZDT/Db2 batch error messages.

Z Data Tools contains four *"flavors"* of Z Data Tools:

- **ZDT/Db2 component** for working with Db2® data.
- Z Data Tools Base component for working with data sets and PDS members, Websphere MQ queues, and z/FS files.
- **ZDT/IMS component** for working with IMS™ data.
- **ZDT/CICS** for working with CICS® resources.

This document describes only the ZDT/Db2 component. For information about the Z Data Tools Base component, ZDT/IMS, and ZDT/CICS components, see these books:

- Z Data Tools User's Guide and Reference
- *Z Data Tools User's Guide and Reference for IMS Data*
- Z Data Tools User's Guide and Reference for CICS

Who should use this document

This document is for three kinds of Db2® users:

- Application programmers who need to test and debug programs
- Service support personnel who need to analyze and fix problems
- System administrators and system operators who need to do routine tasks such as moving large amounts of data

This document assumes that you are familiar with Db2® and with using ISPF.

To use ZDT/Db2 functions in batch jobs, you must already be familiar with JCL.

National characters

Z Data Tools uses the national characters shown in Table 1: National characters on page xiv.

Table 1. National characters

Character	Hexadecimal value	Displayed as, in code pages 37 and 500
Dollar sign	X'5B'	$
Pound sign	X'7B'	#
At sign	X'7C'	@

Note:

1. The dollar sign ($) and the pound sign (#) have special syntactical meaning in Z Data Tools syntax.
2. The at sign (@) is the default value used for the *"Null column input indicator"*.
3. In countries using code pages other than 37 and 500:

Table 1. National characters

(continued)

Character	Hexadecimal value	Displayed as, in code pages 37 and 500
	a. The dollar sign ($), pound sign (#), and at sign (@) as represented on terminal keyboards might generate a different hexadecimal representation, and this might cause an error or unwanted results. For example, in some countries the $ character might generate a X'4A'. b. The hexadecimal values in Table 1: National characters on page xiv might display as different characters to those shown.	

When you enter Z Data Tools commands in batch or online, use the keyboard characters that correspond to the hexadecimal values shown in Table 1: National characters on page xiv.

Prerequisite and related information

ZDT/Db2 documentation supports the following tasks:

Planning for, installing, customizing, and maintaining ZDT/Db2

Refer first to the *Z Data Tools Customization Guide*, which might then refer you to the Z Data Tools Program Directory (included with the product tape) for some information.

Using ZDT/Db2

This document, the *Z Data Tools User's Guide and Reference for DB2 Data*, is a guide to using ZDT/Db2. When using ZDT/Db2 under ISPF, you can also refer to the online help.

How to read syntax diagrams

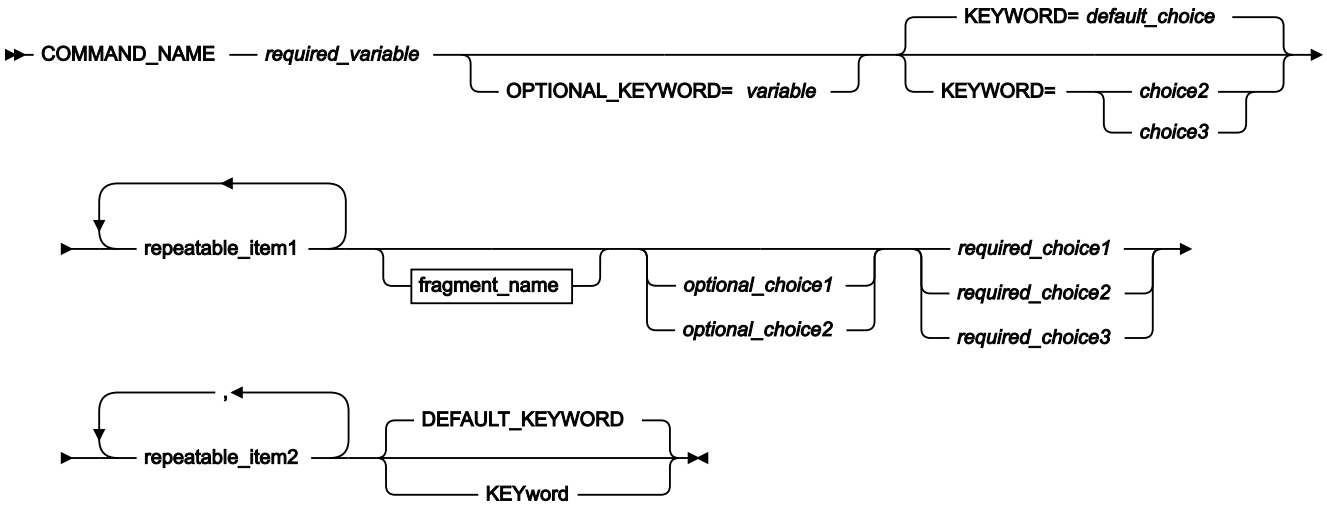
The syntactical structure of commands described in this document is shown by means of syntax diagrams.

Figure 1: Sample syntax diagram on page xvi shows a sample syntax diagram that includes the various notations used to indicate such things as whether:

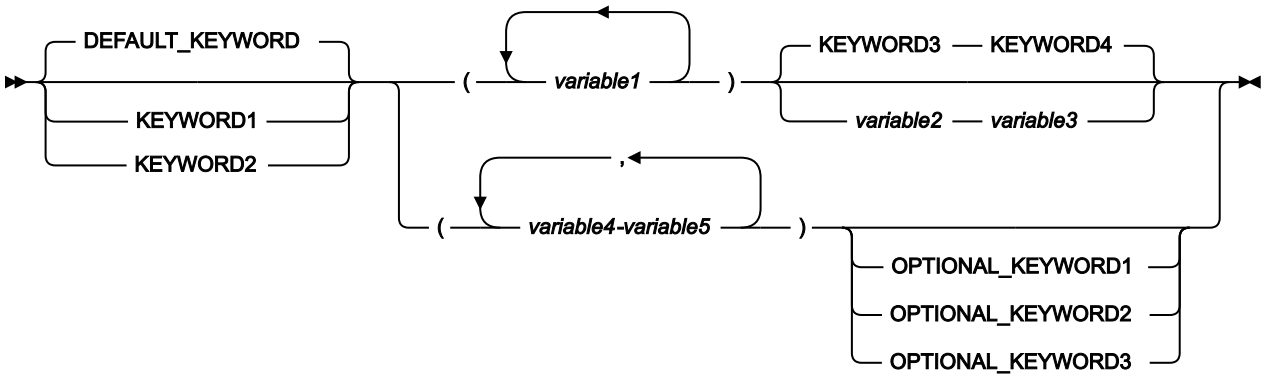
- An item is a keyword or a variable.
- An item is required or optional.
- A choice is available.
- A default applies if you do not specify a value.
- You can repeat an item.

Figure 1. Sample syntax diagram

Figure 1. Syntax



fragment_name



Here are some tips for reading and understanding syntax diagrams:

Order of reading

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

The ►►— symbol indicates the beginning of a statement.

The —► symbol indicates that a statement is continued on the next line.

The ►— symbol indicates that a statement is continued from the previous line.

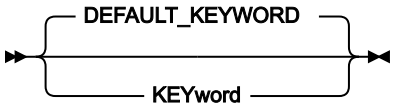
The —►◄ symbol indicates the end of a statement.

Keywords

Keywords appear in uppercase letters.



Sometimes you only need to type the first few letters of a keyword, The required part of the keyword appears in uppercase letters.

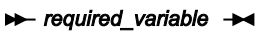


In this example, you could type "KEY", "KEYW", "KEYWO", "KEYWOR" or "KEYWORD".

The abbreviated or whole keyword you enter must be spelled exactly as shown.

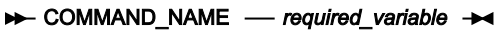
Variables

Variables appear in lowercase letters. They represent user-supplied names or values.



Required items

Required items appear on the horizontal line (the main path).



Optional items

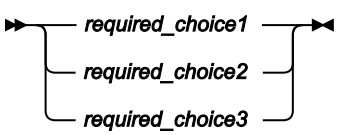
Optional items appear below the main path.



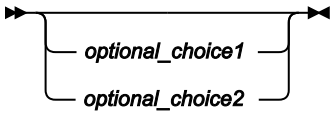
Choice of items

If you can choose from two or more items, they appear vertically, in a stack.

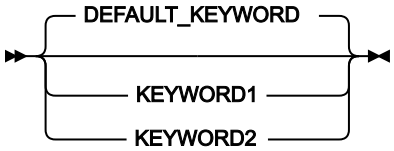
If you *must* choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.

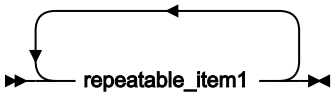


If a default value applies when you do not choose any of the items, the default value appears above the main path.



Repeatable items

An arrow returning to the left above the main line indicates an item that can be repeated.

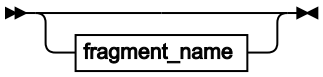


If you need to specify a separator character (such as a comma) between repeatable items, the line with the arrow returning to the left shows the separator character you must specify.

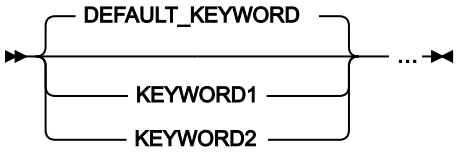


Fragments

Where it makes the syntax diagram easier to read, a section or *fragment* of the syntax is sometimes shown separately.



fragment_name



Summary of changes

These changes, and smaller corrections and additions, are indicated by a "|" change bar in the left margin of the page.

October 2026, changes for V17R1M0

This edition of the document provides information applicable to Z Data Tools Version 17 Release 1 Modification Level 0. Here are the major changes to this document from the previous release:

- Added new topic SHOWPLI primary command on page 1003 to document how to display information about the PL/I compiler currently being used by Z Data Tools.
- Added a new feature to the Z Data Tools User's Guide and Reference that allows users to use the `SHOWPLI` command to identify the PL/I compiler currently used by Z Data Tools. See topic Checking which PL/I compiler you are using on page 29.

Part I. Working with ZDT/Db2

Chapter 1. ZDT/Db2 overview

ZDT/Db2 provides a comprehensive, user-friendly set of tools for working with Db2® data. These tools include the familiar view, edit, copy and print utilities found in ISPF, enhanced to meet the needs of application developers.

This chapter provides an overview of the types of actions you can perform with ZDT/Db2 and detailed concept information about templates, a means of viewing Db2® data in ZDT/Db2.

What you can do with ZDT/Db2

ZDT/Db2 is a powerful set of utility functions for editing, browsing, printing, copying, and maintaining Db2® data. For a description of the Db2® data types that ZDT/Db2 supports, see Db2 data types on page 440.

It also provides utilities for listing Db2® objects, managing Db2® privileges, generating JCL to run Db2® standalone utilities, exporting and importing Db2® tables to or from QSAM or VSAM data sets, creating data to populate Db2® tables, and prototyping SQL SELECT statements.

ZDT/Db2 is an ISPF application program. It uses panels that you can use to select options and to specify parameters, commands and program function (F) keys to simplify requests for common functions, and full-screen format for information display and editing. If data does not fit on the screen, ISPF provides scrolling in all four directions.

Templates

ZDT/Db2 uses *templates* to provide a formatted view of your data, enabling you to view, edit, and manipulate data according to the columns and data types in the table you are working with.

What is a template?

A Z Data Tools/Db2 template is a collection of information that you can use to select and format tables and columns in a Db2® object. If you use a Z Data Tools/Db2 function that interfaces with non-Db2 data, the corresponding template describes the records and fields in the data set.

When you associate a template with data, you can:

Manipulate columns

You can select which columns you want to work with, adjust the order in which they are shown, and modify the column headings.

You can select to show leading zeros for numeric columns. (By default, leading zeros for numeric columns are suppressed. For example, 00057 is displayed as 57.)

Map data

If you are copying, importing, exporting, or loading data, you can *"map"* how data is transferred from the source to the destination.

Select rows

You can specify *row selection criteria* to restrict the data to the rows you want to work with.

Create data

If you are creating a new table, or adding new columns when moving data, you can specify *create attributes* that ZDT/Db2 uses to initialize column values.

If you use a Z Data Tools/Db2 function that uses a template for a Db2® object, you can select one of the following options:

- Specify the name of an existing template
- Use ZDT/Db2 to generate a template (using information from the Db2® catalog for the Db2® object)
- Use the template that was last used with the Db2® object
- Use ZDT/Db2 to generate a template and immediately save the template using a name you specify

Related tasks

Working with templates on page 74

Where you can use templates

You can use templates to view, browse, edit, print, copy, import, export, create, or load data. The following table lists the panels where you can use templates:

Task	Panel
Browse data	View (option B)
View data	View (option 1)
Edit data	Edit (option 2)
Print data	Print (option 3.1)
Copy data	Copy (option 3.3)
Import data	Import (option 3.6)[1]
Export data	Export (option 3.7)[1]
Create data	Create (option 3.8)
Load data	Utilities (option 3.9), Db2® LOAD utility[1]
Unload data	Utilities (option 3.9), Db2® UNLOAD utilities[1]

Related tasks

Viewing and changing Db2 data on page 122
Printing the contents of a Db2 table on page 328
Copying data from one Db2 object to another on page 246

1. You can use a template *or* a copybook with these functions. See Using templates with non-Db2 data on page 77.

Copying data from a VSAM or QSAM file on page 251
Copying data to a VSAM or QSAM file on page 279
Populating a Db2 table with data on page 244
LOAD utility on page 357
UNLOAD (tables) on page 381
UNLOAD (table spaces) on page 386

Related references

Db2 Browse panel on page 612
Db2 View panel on page 631
Db2 Edit panel on page 616
Print Utility panel on page 764
Copy Utility ("From") panel on page 513
Import Utility ("From") panel on page 719
Export Utility ("From") panel on page 698
Data Create Utility panel on page 603
LOAD Utility panel on page 733
UNLOAD Utility (Tables) "From" panel on page 910
UNLOAD Utility (Tables) with LISTDEF panel on page 914
UNLOAD Utility (Table Spaces) panel on page 908
UNLOAD Utility (Table Spaces) with LISTDEF panel on page 909

Chapter 2. System requirements for Z Data Tools for Db2®

ZDT/Db2 supports:

- Db2® Version 13.1 (Function Level 508 and Z Data Tools minimum maintenance level V16.1.4 UO07741)
- Db2® Version 13.1
- Db2® Version 12.1 (Function level 500 or higher is required)

For the complete list of software supported by Z Data Tools, see the topic about Supported software environments in the *Z Data Tools Customization Guide*.

For information about "End of support (EOS)" dates for Db2 for z/OS, refer to the IBM Product lifecycle website.

Where variations exist within ZDT/Db2 depending on the version of Db2® you are using, these variations are highlighted in this document by indicators that identify the start and end of a section of text. For example:

Note: Some examples shown in this document might reference older versions of DB2®. If you are using a later version of Db2® you might find minor variations in the appearance of ZDT/Db2 panels.

Chapter 3. Getting started with ZDT/Db2

This chapter provides answers to questions such as: "How do I enter and exit the application?", "How do I move around the application?", "How do I get help?", and "What do I need to do before I start working?" .

Related tasks

Starting and exiting ZDT/Db2 on page 25
Selecting the Db2 subsystem when more than one is available on page 30
Using the ZDT/Db2 interface on page 35
Getting help on page 57
Setting default processing options on page 62

Starting and exiting ZDT/Db2

ZDT/Db2 is an ISPF-developed application and is usually run from within an ISPF session. Exiting from ZDT/Db2 returns you to your ISPF session.

Related tasks

Starting ZDT/Db2 on page 26
Exiting from ZDT/Db2 on page 29

TSO region size

The minimum supported TSO region size for ZDT/Db2 is 16MB. While ZDT/Db2 can operate with region sizes in the range 4MB - 16MB, these region size specifications restrict the amount of *"below the line"* storage. If you encounter storage-related abends running with TSO region sizes less than 16 MB, attributable to *"below the line"* storage shortages, the remedy is to increase the TSO region size.

When processing large Db2® objects, significant amounts of 31-bit storage may be required.

For most applications a TSO region size of 32MB is a suitable starting point, this should be increased if there is a need to process large Db2® objects.

When using the GEN prefix command, the minimum suggested TSO region size is 32MB. Increase this to 64MB when processing large databases.

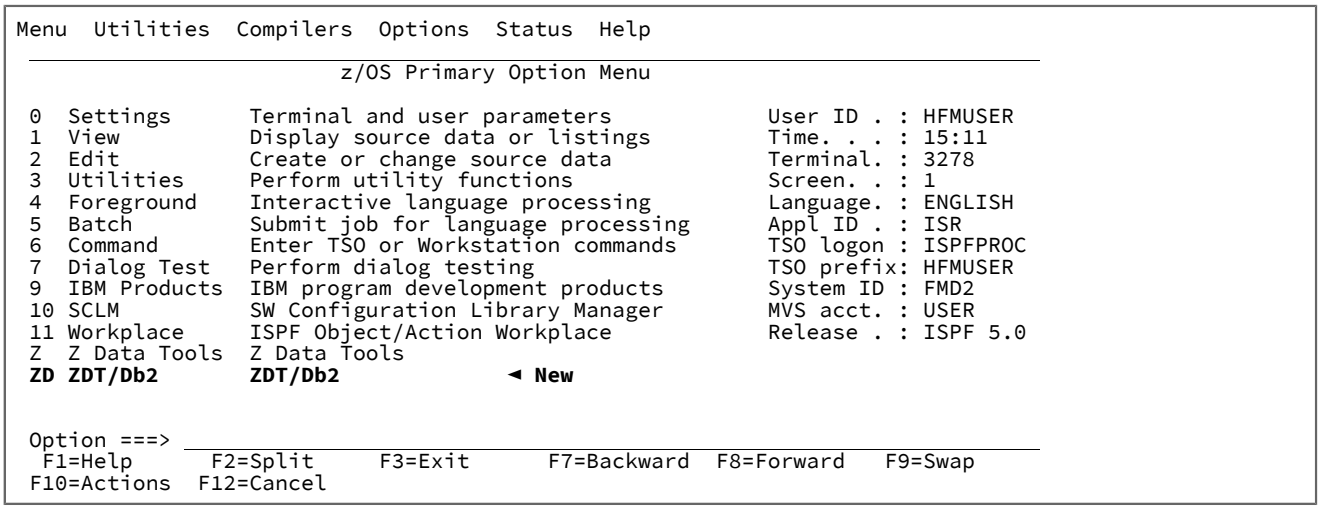
Related tasks

Storage considerations on page 138
Viewing large tables or views on page 139

Starting ZDT/Db2

How you start ZDT/Db2 depends on how it has been installed on your system. Typically, ZDT/Db2 is installed as an option on your z/OS® Primary Option Menu:

Figure 3. z/OS® Primary Option Menu panel showing ZDT/Db2 option



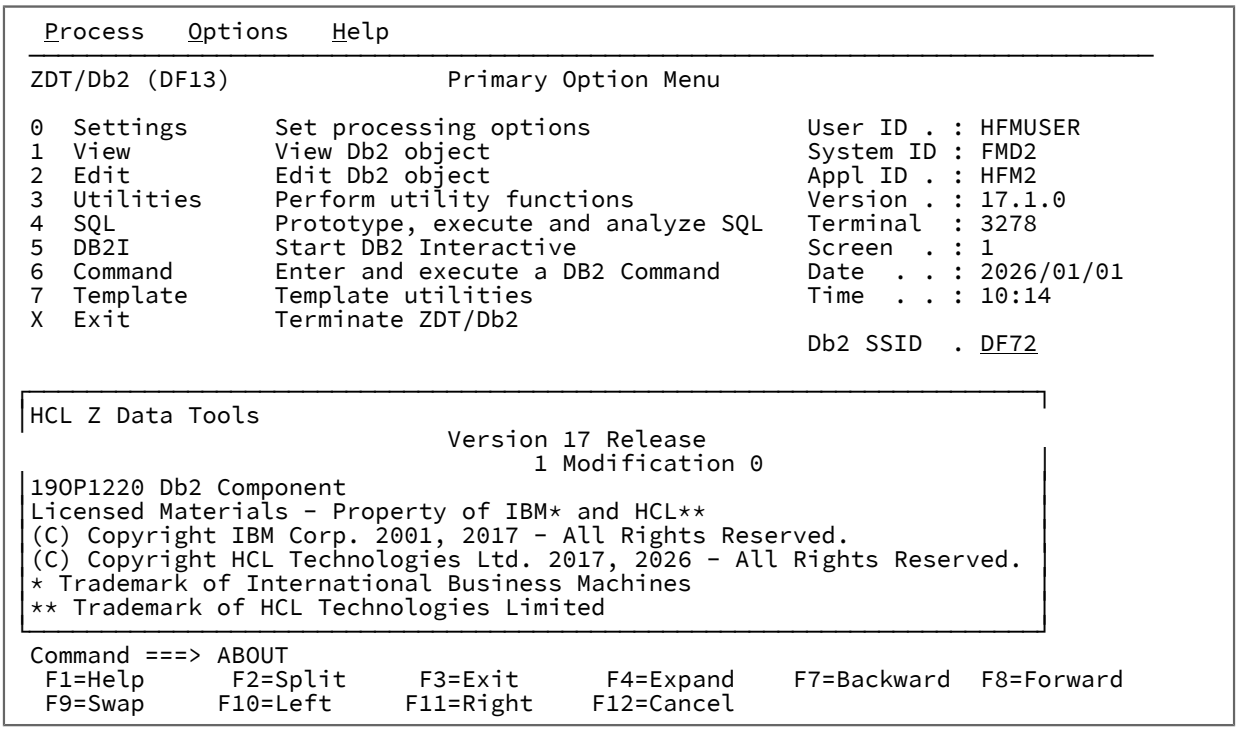
To start ZDT/Db2:

1. Select the ZDT/Db2 option from the z/OS® Primary Option Menu.

   Note: If ZDT/Db2 is not an option on your menu, ask your systems administrator to explain the startup process used at your site.

   When you start ZDT/Db2, the ZDT/Db2 Primary Option Menu is shown, with the current version information displayed in a pop-up message box.

Figure 4. ZDT/Db2 Primary Option Menu panel with version information

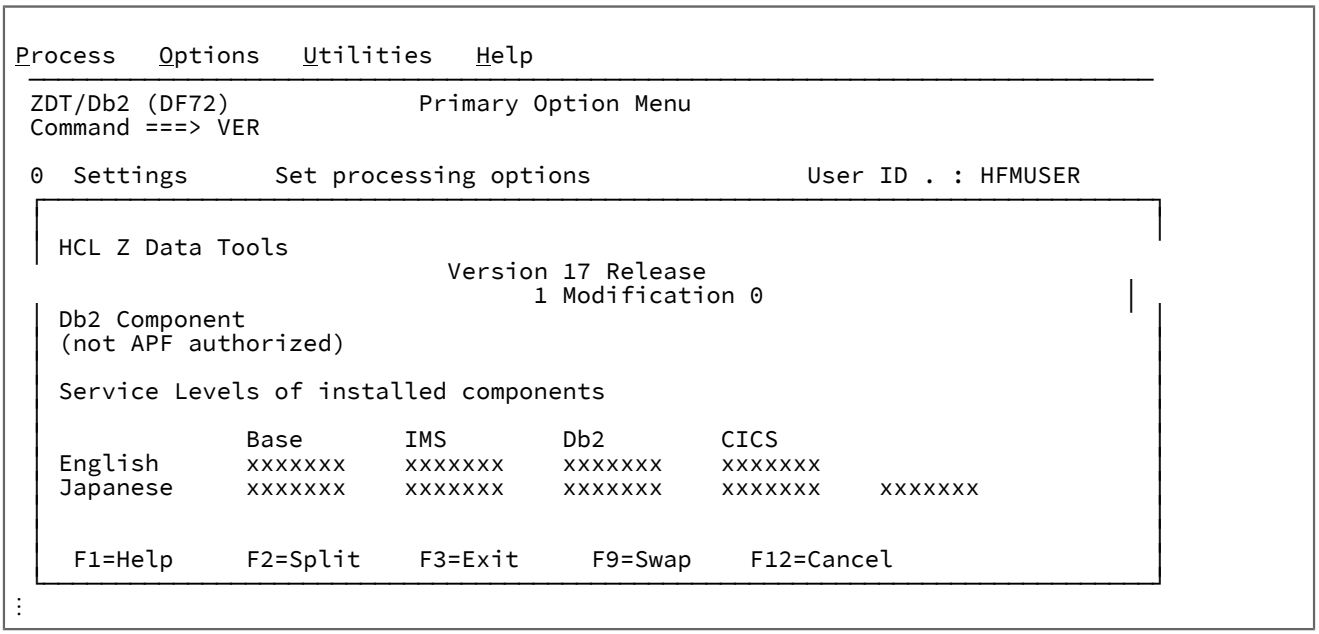


2. Press Enter to close the version window.

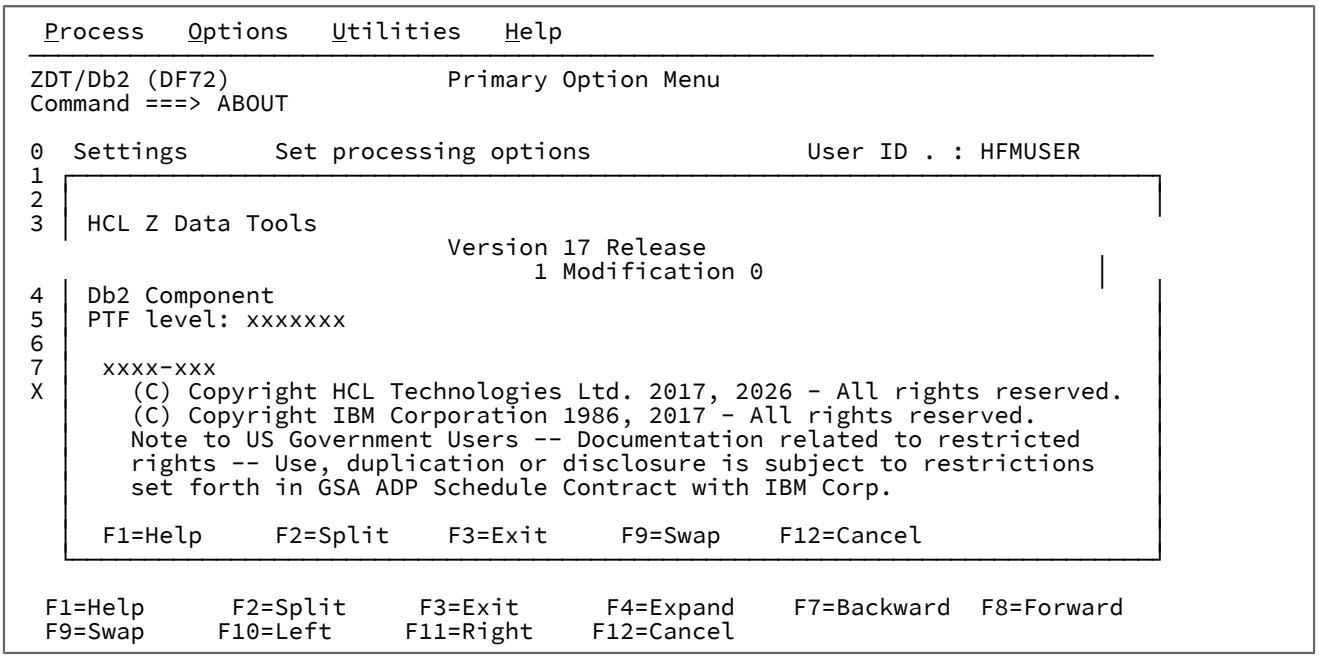
Checking your ZDT/Db2 version

To display the full version information for ZDT/Db2, enter VER on the command line of any panel. The current ZDT/Db2 version number and the PTF number for each ZDT/Db2 component is displayed in a window. The pop-up also indicates whether or not ZDT/Db2 is APF-authorized.

Figure 5. ZDT/Db2 Primary Option Menu panel with PTF information



An alternative way to display version information is to select **Help > 7. About** from the Action Bar. The current ZDT/Db2 version number and the PTF number of the Db2® component are displayed. The pop-up panel also shows the copyright information and any notes that are shipped with the product.



Related references

VER primary command on page 1010
Action bar pull-down menu on page 445

Checking which COBOL compiler you're using

ZDT/Db2 uses the currently loaded COBOL compiler to create a template from a COBOL copybook.

To display details about the current COBOL compiler, enter SHOWCOB on the Command line of any panel. ZDT/Db2 displays details about the current COBOL compiler in a window.

Related topics

SHOWCOB primary command on page 1003

Exiting from ZDT/Db2

You can exit from ZDT/Db2 from the Primary Options Menu panel in any of the following ways:

- Press the Exit function key (F3).
- Enter `X` (or `EXIT` or `END`) on the command line.
- Select **Process > Exit ZDT/Db2** from the Action Bar.

To exit the application from a panel within ZDT/Db2:

- Enter `=X` on the command line.

Related references

END primary command on page 942
Action bar pull-down menu on page 445

Checking which PL/I compiler you are using

Z Data Tools uses the currently loaded PL/I compiler to create a template from a PL/I copybook.

To display information about the currently loaded PL/I compiler, enter SHOWPLI on the command line of any panel. File Manager displays information about the PL/I compiler that is currently in use.

Related information

SHOWPLI primary command on page 1003

Selecting the Db2® subsystem when more than one is available

On most ZDT/Db2 panels, the currently connected Db2® subsystem is displayed immediately after the product name.

For instance, in Figure 6: Primary Option Menu panel on page 31, "`ZDT/Db2 (DFG2)`" (before the panel title) shows that the current Db2® subsystem is DFG2.

If your system contains more than one active Db2® subsystem, you can select the subsystem to which you want ZDT/Db2 to connect by entering a value in the **Db2 SSID** field on the Primary Option Menu. You can also use the SSID command in some ZDT/Db2 panels.

The first time you use ZDT/Db2, if your system contains multiple active Db2® subsystems, ZDT/Db2 immediately displays the Db2® Subsystem Selection panel.

Note:

1. ZDT/Db2 can only run on a system that has at least one active Db2® subsystem that uses a supported version of Db2®.
2. If your system contains only one active Db2® subsystem, ZDT/Db2 automatically connects to that subsystem.

Related tasks

Selecting the Db2 subsystem by using the Db2 SSID field on page 30
Selecting the Db2 subsystem by entering the SSID command on page 32

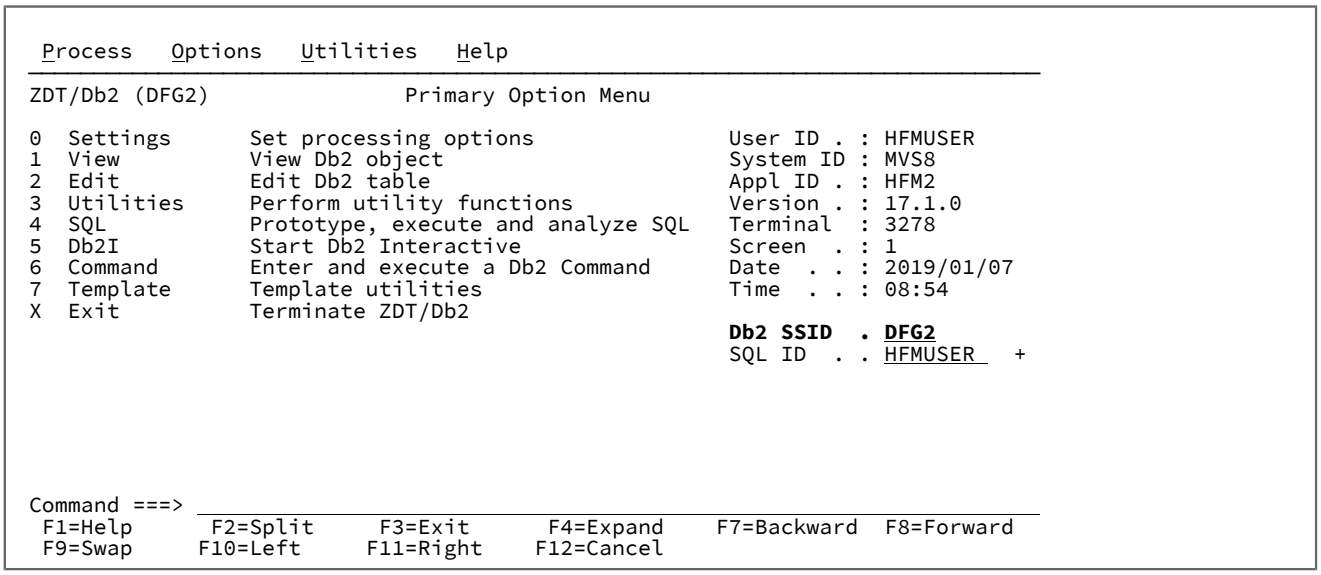
Related references

Primary Option Menu panel on page 760
Db2 Subsystem Selection panel on page 624

Selecting the Db2® subsystem by using the Db2 SSID field

When you start ZDT/Db2, the Primary Option Menu panel is displayed.

Figure 6. Primary Option Menu panel



The **Db2 SSID** field shows the ID of the Db2® subsystem to which ZDT/Db2 is currently connected. (The Db2® subsystem is also shown before the panel title.) In Figure 6: Primary Option Menu panel on page 31, ZDT/Db2 is connected to the DFG2 subsystem.

To select a Db2® subsystem using the **Db2 SSID** field, you can use any of the following methods:

- Overtype the SSID that is currently displayed with the ID of another active Db2® subsystem, and then press Enter.

  For example, in Figure 6: Primary Option Menu panel on page 31, to connect to the Db2® subsystem DFB2, overtype DFG2 with `DFB2`, and then press Enter.
- Enter an asterisk (*) or blanks in the **Db2 SSID** field, and then press Enter. ZDT/Db2 displays the Db2 Subsystem Selection panel showing a list of all the Db2® subsystems in your environment from which you can select the Db2® subsystem you want.
- Enter a partial ID, starting or ending with an asterisk (*), or enclosed within two asterisks, in the **Db2 SSID** field, and then press Enter. ZDT/Db2 displays the Db2 Subsystem Selection panel showing a restricted list of Db2® subsystems from which you can select the Db2® subsystem you want.

  The following examples demonstrate how you can use the (*) wildcard if specifying a partial Db2® subsystem ID:

  DB*

  Lists all Db2® subsystems whose ID starts with *"DB"*

  ***TST**

  Lists all Db2® subsystems whose ID ends with *"TST"*

  2

  Lists all Db2® subsystems whose ID contains *"2"*

Note: To change the Db2® subsystem from a panel other than the Primary Option Menu panel, use the SSID command.

Related tasks

Using the Db2 Subsystem Selection menu on page 33
Selecting the Db2 subsystem by entering the SSID command on page 32

Related references

Primary Option Menu panel on page 760
Db2 Subsystem Selection panel on page 624

Selecting the Db2® subsystem by entering the SSID command

To connect to a different Db2® subsystem, use the SSID command followed by the name of the Db2® subsystem to which you want to connect.

The SSID command can be used from menu panels, option panels, and most function entry panels. For example, to change to the DB27 subsystem, enter the command:

```
SSID DB27
```

When you use the SSID command, be aware of the following restrictions:

- The name you specify must be an active Db2® subsystem.
- The name you specify must not contain an asterisk (*) or wildcard.
- Once a Z Data Tools/Db2 dialog has started, you cannot use the SSID command to change the currently connected Db2® subsystem.

To display a list of all the Db2® subsystems in your environment, use the **Db2 SSID** field on the Primary Option Menu panel.

Note:

- DB2SYS is a synonym for the SSID command. When ZDT/Db2 was installed these command names might have been customized to something else. If so, references to the SSID command in this document will apply instead to these customized command names.

  To verify the names of the SSID commands at your site, refer to the Db2 Subsystem Selection Tutorial Help panel:



1. From the Primary Option Menu panel, press the Help function key (F1).
2. From the *"ZDT/Db2 Tutorial - Table of Contents"* panel, select item **D** (Db2 SSID).

   The *"ZDT/Db2 Tutorial - Db2 Subsystem Selection"* panel shows the name of the command and its synonym that were set when ZDT/Db2 was installed.

- You can abbreviate the command name to as few as 4 characters.

Related tasks

Using the Db2 Subsystem Selection menu on page 33
Selecting the Db2 subsystem by using the Db2 SSID field on page 30
Accessing Tutorial Help on page 57

Related references

Primary Option Menu panel on page 760

Using the Db2® Subsystem Selection menu

You use the Db2® Subsystem Selection menu to select the Db2® subsystem to which you want ZDT/Db2 to connect. It shows a list of the Db2® subsystems in your environment and is displayed:

- The first time you use ZDT/Db2 (providing your system contains multiple active Db2® subsystems),
- If you enter any of the following in the **Db2® SSID** field on the Primary Option Menu:
  - An asterisk (*)
  - Spaces
  - A partial ID, starting or ending with an asterisk (*), or both

If you enter a partial ID in the **Db2® SSID** field on the Primary Option Menu, the list of Db2® subsystems is restricted accordingly.

Related references

Primary Option Menu panel on page 760
Db2 Subsystem Selection panel on page 624

Selecting a subsystem

To select a Db2® subsystem from the Db2® Subsystem Selection menu:

1. Enter `s` (or ?) in the **Sel** column for the Db2® subsystem you want.

   The status of the subsystem you select must be "ACTIVE" or "GROUP", and the subsystem must have been defined in the ZDT/Db2 installation module. For Db2® systems that do not meet these criteria, the selection field is disabled.

2. Press Enter.

   ZDT/Db2 displays the Primary Option Menu. The ID of the subsystem you selected is displayed in the **Db2® SSID** field showing that ZDT/Db2 is now connected to that Db2® subsystem.

Related references

Primary Option Menu panel on page 760
Db2 Subsystem Selection panel on page 624

Refreshing the list of Db2® subsystems

To refresh the list of Db2® subsystems on the Db2® Subsystem Selection menu, use the Refresh (F5) function key or enter the primary command REFRESH.

The refreshed list of Db2® subsystems reflects any changes that have occurred since you last displayed the list (for example, the status of each Db2® subsystem).

Related references

REFRESH primary command on page 993
Db2 Subsystem Selection panel on page 624

Restricting the list of Db2® subsystems

You can restrict the list of Db2® subsystems shown on the Db2® Subsystem Selection menu by using the SHOW primary command or the Show function key (F6).

Either of the following commands lists all Db2® subsystems:

```
SHOW *

SHOW
```

The following examples demonstrate how you can use the SHOW primary command to restrict the list of Db2® subsystems shown on the Db2® Subsystem Selection menu using the asterisk (*) wildcard and a partial Db2® subsystem ID:

SHOW MM*

Restricts the list to Db2® subsystems with an ID starting with *"MM"*

SHOW *PBJ

Restricts the list to Db2® subsystems with an ID ending with *"PBJ"*

SHOW *V*

Restricts the list to Db2® subsystems with an ID containing *"V"*

The Show function key (F6) works in the same way as the SHOW command, except that it uses the contents of the command line as the selection criteria. For example:

- If the command line is blank and you press the Show function key (F6), ZDT/Db2 lists all Db2® subsystems. (You would get the same result if the command line contained an asterisk (*)).
- If you type `DBT*` on the command line and press the Show function key (F6), ZDT/Db2 lists all Db2® subsystems whose ID begins with “DBT”.
- If you type `*26` on the command line and press the Show function key (F6), ZDT/Db2 lists all Db2® subsystems whose ID ends with “26”.
- If you type `*AB*` on the command line and press the Show function key (F6), ZDT/Db2 lists all Db2® subsystems whose ID contains “AB”.

Related references

Db2 Subsystem Selection panel on page 624
SHOW primary command on page 1001

Using the ZDT/Db2 interface

The ZDT/Db2 interface is based upon the ISPF model. Tasks are performed by processing a function, together with its applicable parameters. Panels provide a user-friendly way of selecting a function and supplying the parameter information. Some functions can be processed in your choice of "foreground" or "batch" mode. In foreground mode, the selected function is processed immediately and any results are returned to you on screen or directed to print outputs, as dictated by your default settings. In batch mode, JCL is generated from the panel information, and then presented to you for editing. You can modify this code and submit it to a processing queue.

This section provides instructions on how to navigate to panels within ZDT/Db2 and how to specify common panel information.

Navigating in ZDT/Db2

You navigate around ZDT/Db2 panels in the same way as any other panels under ISPF. That is, you launch processing panels from menu panels, by entering the appropriate menu option number on the command line. ZDT/Db2 uses a Primary Options Menu panel to provide access to processing panels that are related to a particular function or group of functions. In some cases, an option on the Primary Options menu leads to another menu panel, from which you can select the required processing panel.

To access a panel, choose one of the following methods:

- Type the menu option number on the Command line and press Enter. Repeat for each nested menu until the required processing panel is displayed.
- From the Primary Option Menu, type the complete menu path to the required panel, separating each menu level with a period, then press Enter. For example, entering `3.4` takes you directly to the Object List Utility panel.
- Use "point-and-shoot" for fields on menu panels. To use this facility, place the cursor on the name of the menu option that is required, and press Enter.

Note:
ISPF provides an "=" parameter that you can specify before an option selection. For example, `=3.4`. When supported, the "=" parameter enables direct navigation to the specified menu option from another unrelated option within an ISPF application.

The "=" parameter is not supported by ZDT/Db2.

To exit from a panel, choose one of the following methods:

- Press the Exit function key (F3).
- Type `X` on the command line and press Enter.
- Type `END` on the command line and press Enter.

Many ZDT/Db2 panels have more information than can be seen in a single screen, particularly when you are running in 80x24 display. In this case, a *"More"* indicator is displayed near the upper right corner of the panel with a plus sign (+), a minus sign (-), or both.

Figure 7. Example panel showing "More" message

```
                    ZDT/Db2 Tutorial - Ed/Br Options
                                                                More:   - +
  session are only presented to Db2 when an explicit save command is issued,
  or at the end of the edit session.

Save data
  When this option is selected, all pending changes are submitted to Db2 for
  processing whenever the enter key is pressed (and no other operations are
```

A plus sign (+) adjacent to the word *"More"* indicates more information is available on a following panel. Press the NxtPage function key (F11) to display the next information panel.

A minus sign (-) adjacent to the word *"More"* indicates more information is available on a previous panel. Press the PrvPage function key (F10) to display the previous information panel.

To scroll down the panel, choose one of these methods:

- Press the Forward function key (F8). This scrolls down one screenful of lines.
- Enter `FORWARD` or `DOWN` on the command line. These commands scroll down one screenful of lines.
- Enter `BOTTOM` on the command line. This takes you to the bottom of the panel.

To scroll up the panel, choose one of these methods:

- Press the Backward function key (F7). This scrolls up one screenful of lines.
- Enter `BACKWARD` or `UP` on the command line. These commands scroll up one screenful of lines.
- Enter `TOP` on the command line. This takes you to the beginning of the panel.

Related references

Scrolling on page 160
BACKWARD primary command on page 923
BOTTOM primary command on page 924

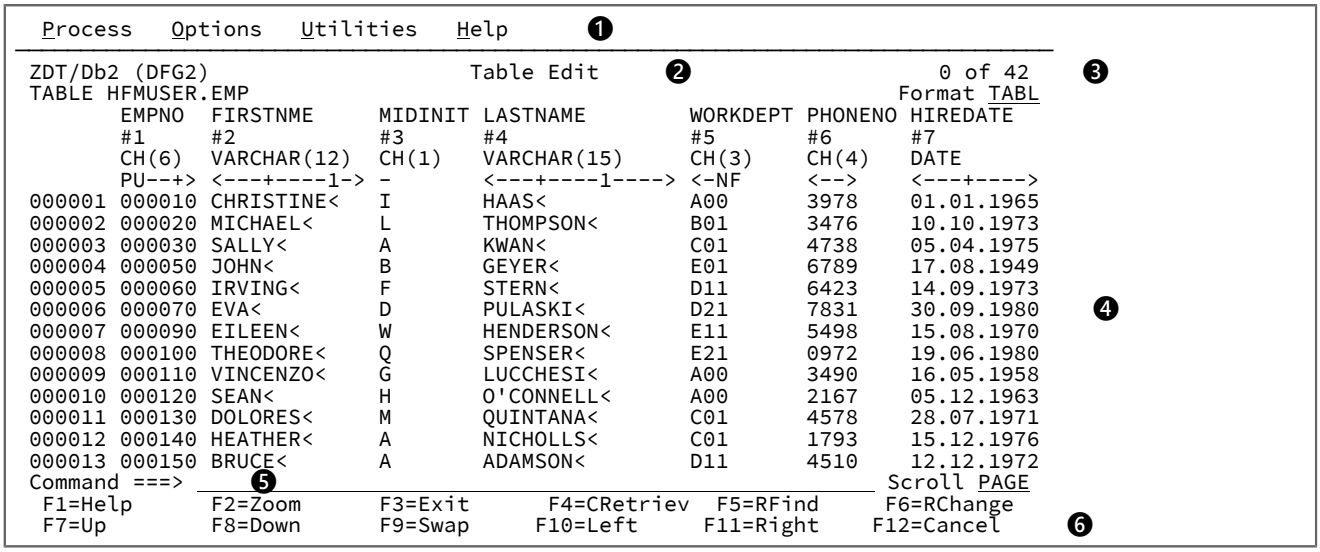
DOWN primary command on page 939
END primary command on page 942
FORWARD primary command on page 963
TOP primary command on page 1007
UP primary command on page 1009
Screen size on page 40

ZDT/Db2 panel features

This section describes the general features of ZDT/Db2 panels.

Figure 8: Typical ZDT/Db2 panel on page 38 shows a typical ZDT/Db2 panel.

Figure 8. Typical ZDT/Db2 panel



In general, the following lines are displayed on ZDT/Db2 panels:

1 Action bar

Provides access to pull-down menus that give you a fast way to move around the product.

Note: The action bar is not displayed if you select Start Db2® Interactive (option 5).

2 Panel title

Identifies the function being carried out.

❸ **Row count information (editor sessions only)**

Shows row number (of top row currently displayed) and total number of rows fetched for the current ZDT/Db2 editor session.

❹ **Panel body**

Shows information pertaining to the panel.

❺ **Command line**

Use to enter a command or, on a menu, to enter either a command or an option. The command can be any ISPF command or valid ZDT/Db2 primary command for the ZDT/Db2 function being carried out.

To change the position of the command line (the bottom or top of the panel, see the *ISPF User's Guide*.

❻ **Function keys**

Displays the settings for the function keys (also known as *"Programmable Function"* or *"PF"* keys). For most ZDT/Db2 panels, these settings are the ISPF function key values. However, for some ZDT/Db2 panels, these settings are preset to correspond with the needed values for the ZDT/Db2 function.

For information about default function key values, and how to define your own function key values, see the *ISPF User's Guide*.

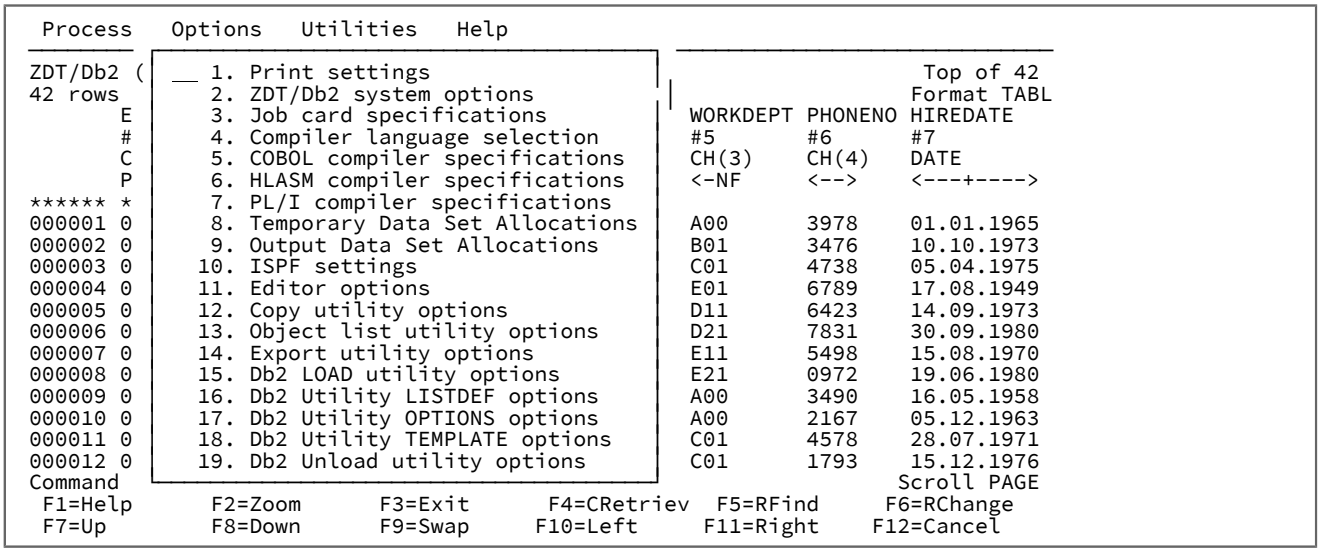
Selecting an item from the action bar

To select an item from the action bar:

1. Move the cursor to the item on the action bar; for example, **Options** (see Figure 9: Action bar menu on page 40).
2. Press Enter. ZDT/Db2 displays a pull-down menu with choices relating to the action bar item you selected.
3. Select a choice from the pull-down menu. Either:
   - Type the pull-down menu selection number (for example, `6` for **Print settings**), and press Enter, or
   - Move the cursor with the Up or Down arrow keys to the required choice, and press Enter.

An asterisk in front of a choice on a menu indicates that the choice is not available for the current panel.

Figure 9. Action bar menu



Related references

Action bar pull-down menu on page 445

Screen size

ZDT/Db2 is designed to handle screen sizes larger than 24 x 80 characters.

For information about ISPF settings affecting terminal characteristics, see the *ISPF User's Guide*.

Scrollable input and display fields for long names

ZDT/Db2 uses scrollable fields on ZDT/Db2 panels to allow you to input and view values longer than the field length.

"There are two types of scrollable fields on ZDT/Db2 panels:

Input fields

The fields on ZDT/Db2 panels where you can *enter* information, such as the name of an owner (or creator), the name of a Db2® object, or the name of a Db2® column.

Display fields

The fields on ZDT/Db2 panels that *display* information as a result of some action you have taken.

"The following two sections describe how to work with both types of scrollable fields."

Working with scrollable input fields

On panels where you can enter the name of:

- The owner (or creator),
- A Db2® object, or
- A column

ZDT/Db2 provides scrollable input fields that allow you to enter values longer than length of the displayed entry field.

You can recognise a scrollable field by the presence of one of the following adjacent to the end of the entry field:

+

Plus sign. Indicates the field is scrollable to the right

-

Minus sign. Indicates the field is scrollable to the left

+-

Plus sign and minus sign. Indicates the field is scrollable to the right or left

If the length of the input field displayed on the panel is insufficient for the value you want to enter, perform one of these actions:

- Progressively scroll to the right as you enter the value with the Right function key (F11) or by entering the RIGHT primary command (with the cursor positioned within the entry field),
- Press the Expand function key (F4), or enter the EXPAND primary command, to display a window that allows you to enter the full length of the value

Figure 10: Db2 View panel: format displayed when connected to a Db2 V8 system on page 42 shows the format of the Db2® View panel. The *"+"* symbol adjacent to the **Owner** and the **Name** input fields indicates that both are scrollable fields.

Figure 10. Db2® View panel: format displayed when connected to a Db2® V8 system

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Db2 View

Specify the Db2 Object:
   Location  . . . . . _________________          Database . . ________ (optional)
   Owner . . . . . . . _________________  +       Table space  ________ (optional)
   Name  . . . . . . . ____________________________________________________  +

   Row count . . . . . 100_____      Number of rows to browse

Template:
   Data set name . . . 'HFMUSER.TEMPLATE'_____________________________
   Member  . . . . . . SALARY__

Processing Options:
   Template usage                    Enter "/", "A"lways to select option
   3  1. Above                       _  Edit options
      2. Previous                    _  Edit template
      3. Generate from table         _  Re-edit template
      4. Generate/Replace


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Actions   F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

The **Owner** input field displayed on the panel allows for up to 16 characters to be entered. To enter the name of an owner that is longer than 16 characters, you must use the scrolling or expand capabilities of the **Owner** input field.

For example, to enter the 21-character name *"HANSCHRISTIANANDERSON"*, you can perform either of these actions:

- Type the first 16 characters of the name (`HANSCHRISTIANAND`) in the **Owner** field, use the left arrow key (◄) to position the cursor back in the **Owner** input field, press the Right function key (F11) to scroll right, and then type the remainder of the name (`ERSON`) in the **Owner** field,
- Press the Expand function key (F4), or enter the EXPAND primary command, to display a window that allows you to enter the full name, and then press the Exit function key (F3) to return to the Db2® View panel.

  Figure 11: Window when entering a long input name on page 42 shows the window for the Owner field with the full name entered.

  Figure 11. Window when entering a long input name

```
┌──────────────────────────────── HFM2ECRE+0 ────────────────────────────────┐
│                                                        Line   1 of      2 │
│ HANSCHRISTIANANDERSON                                                     │
│ ____________________________________________                              │
│                                                                           │
│                                                                           │
│                                                                           │
│                                                                           │
│                                                                           │
│                                                                           │
│                                                                           │
│ Command ===>                                              Scroll ===> PAGE│
│  F1=Help     F2=Split    F3=Exit     F5=Rfind    F6=Rchange  F7=Up       │
│  F8=Down     F9=Swap    F10=Left    F11=Right   F12=Cancel                 │
└───────────────────────────────────────────────────────────────────────────┘
```

When you press the Exit function key (F3) to return to the Db2® View panel, the **Owner** input field shows the first 16 characters of the name you entered:

```
⋮
 Specify the Db2 Object:
    Location  . . . . . _________________          Database . . _________ (optional)
    Owner . . . . . . . HANSCHRISTIANAND  +        Table space  _________ (optional)
    Name  . . . . . . . ____________________________________________________  +
⋮
```

To see the rest of the name in the **Owner** input field, scroll right by either pressing the Right function key (F11) or entering the RIGHT primary command (ensuring the cursor is positioned within the **Owner** input field):

```
⋮
 Specify the Db2 Object:
    Location  . . . . . _________________          Database . . _________ (optional)
    Owner . . . . . . . ERSON             +        Table space  _________ (optional)
    Name  . . . . . . . ____________________________________________________  +
⋮
```

Deleting data in scrollable input fields

If you want to delete all the data in a field that is displayed as a scrollable input field, take care to ensure that you do, in fact, delete all the data (and not just the displayed data).

If you use the field delete key combination to delete data, ZDT/Db2 deletes only the data that is currently displayed. If the field contains more data than is displayed, the undisplayed data is left undeleted.

It is recommended that, before deleting data in a field that is displayed as a scrollable field, you first display the entire contents of the field by pressing the Expand function key (F4) or entering the EXPAND primary command. You can then use the field delete key combination to delete all data in the field at once.

Working with scrollable display fields

Many ZDT/Db2 panels show data displayed in scrollable fields. ZDT/Db2 uses such fields where the length of the field containing the data to be displayed is longer than the field displayed on the panel.

You can recognise a scrollable field by the presence of one of the following adjacent to the end of the display field:

+

Plus sign. Indicates the field is scrollable to the right

-

Minus sign. Indicates the field is scrollable to the left

+-

Plus sign and minus sign. Indicates the field is scrollable to the right or left

If the data to be displayed is longer than the length of the display field on the panel, you can view the rest of the data by performing either of these actions:

- Progressively scrolling to the right by pressing the Right function key (F11) or by entering the RIGHT primary command (with the cursor positioned within the display field),
- Pressing the Expand function key (F4) or entering the EXPAND primary command to display a window that allows you to view the entire contents of the field.

Figure 12: Table Details panel: example showing scrollable fields on page 44 shows the Table Details panel which includes a number of scrollable display fields (those with a *"+"* symbol adjacent to the end of the field: **Table owner**, **Table name**, **Created by**, **Created**, and **Altered**). Scrollable display fields indicate that, possibly, not all of the data held in the field is being displayed.

Note: The presence of a *"+"* symbol indicates that the field is scrollable and *may* contain more data than is currently being displayed. It does not necessarily mean that more data exists than is currently being shown.

Figure 12. Table Details panel: example showing scrollable fields

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Table Details
                                                                More:     +

   Details about table (label) : KPS8610LO12345678901234567890123456789012345678

General Data:
   Table owner . . . . : KPS8610LO  +  Table name  . . . . : DUMMY_TABLE_WI  +
   Created by  . . . . : SHRIKES    +  Database name . . . : DSNDB04
   Table space name  . : DUMMYRTA      DB ID for database  : 4
   Object ID for table : 3             Primary key OBID  . : N/A
   Maximum row length  : 69            Primary key columns : N/A
   Number of columns . : 4             Edit procedure name : N/A
   Validation proc.  . : N/A           Check pending flag  : No
   Parent relations  . : 0             Check constraints . : None
   Child relations . . : 0             Encoding scheme . . : E - EBCDIC
   Auditing  . . . . . : AUDIT NONE    Status  . . . . . . : No primary key
   Data capture  . . . : No            Created . . . . . . : 2004-05-10-11.  +
   Restrict on DROP  . : No            Altered . . . . . . : 2004-05-10-11.  +
   Release created . . : L             Dep. query tables . : 0
Command ===>
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Actions   F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

For example, to see more of the data in the **Created** display field (in the bottom righthand part of the Table Details panel), you can perform either of these actions:

- Scroll right by either pressing the Right function key (F11) or entering the RIGHT primary command (ensuring the cursor is positioned within the **Created** field).

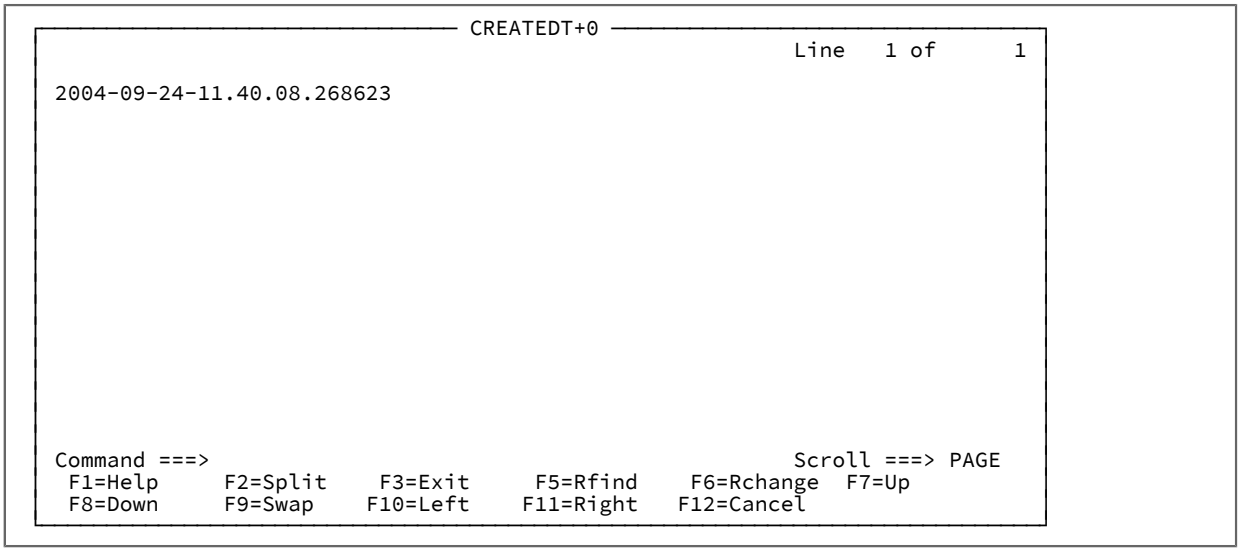
  The **Created** display field now shows more data:

  ```
  ⋮
     Data capture  . . . : No            Created . . . . . . . : 1.40.08.268623  -
  ⋮
  ```

  Note that the scroll symbol has changed from a *"+"* to a *"-"*, indicating that, in this case, you have scrolled to end of the field.

- Press the Expand function key (F4) or enter the EXPAND primary command to display a window that shows the entire contents of the **Created** display field:

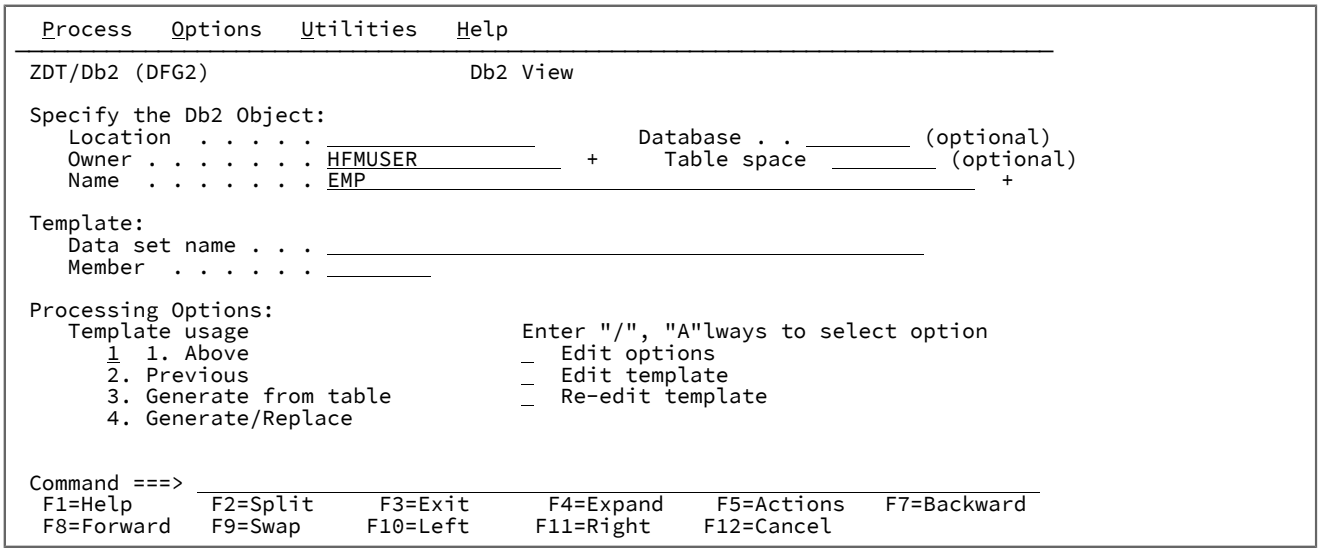
  Figure 13. Window showing entire contents of display field



Selecting options on ZDT/Db2 panels

There are several ZDT/Db2 panels, such as the Db2® View panel shown in Figure 14: ZDT/Db2 panel showing selectable options on page 45, where you can select one or more options for the ZDT/Db2 function you are using.

Figure 14. ZDT/Db2 panel showing selectable options



The options on ZDT/Db2 panels are displayed in two different formats:

- A list of numbered options in the form of a menu.

  To select one of these options, type the option number in the entry field provided. For example, in the Db2® View panel shown in Figure 14: ZDT/Db2 panel showing selectable options on page 45, to select **Previous**, type `2`.

- One or more options that you can select or deselect.

  To select one of these options for the current session of the ZDT/Db2 function only, type a *"/"* in the entry field provided. To deselect the option, type a space in the entry field.

  If the panel shows you can select the option by entering a *"/"* or an *"A"* (such as the Db2® View panel):
  - To select an option for the current session of the ZDT/Db2 function only, type a "/".
  - To retain this option as selected from one session of the ZDT/Db2 function to the next, type an "A". Use this selection character if you want ZDT/Db2 to *always* select this option before continuing processing the ZDT/Db2 function.

    For example, if you always want to edit the template before viewing data, select **Edit template** on the Db2® View panel using the selection character "A".

You must select all the required options before you press Enter.

Related references

Db2 View panel on page 631

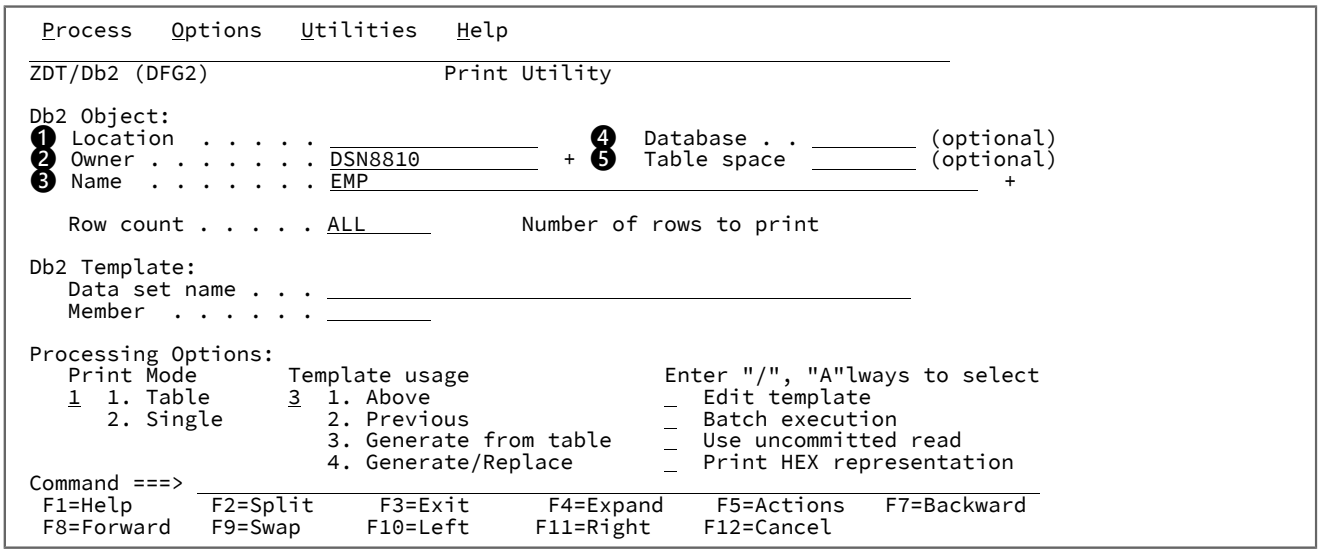
Specifying a Db2® object name

When you use any of the ZDT/Db2 functions listed below, you specify the name of the Db2® table or view you want to work with in the same way.

- View (option 1)
- Edit (option 2)
- Print (option 3.1)
- Copy (option 3.3)
- Import (option 3.6)
- Export (option 3.7)
- Create (option 3.8)
- Prototyping: Basic (option 4.1)
- Browse (B command on Primary Option Menu panel, or B prefix command against an object in a list of Db2® objects)

Figure 15: Print utility panel on page 47 shows a typical panel containing the entry fields for specifying a Db2® object,

Figure 15. Print utility panel



The panels for each of the preceding ZDT/Db2 functions contain the following entry fields, all of which are optional apart from **Name**:

❶ Location

The location of the table or view.

If the table or view that you want to work with is at your current location, leave this field blank. Otherwise, type:

- The full name of the remote location
- An asterisk (*) to display all available remote locations, or
- A pattern using wild card characters

If you specify an asterisk on its own or a pattern, ZDT/Db2 displays a selection list showing the locations that match the input specification. To select the location you want, type `s` against the required entry and press Enter. The location selection list is not available unless access to certain Db2® catalog tables has been granted to ZDT/Db2 users. When the access is not available, you can still access a remote Db2® system by spe cifying the exact location name.

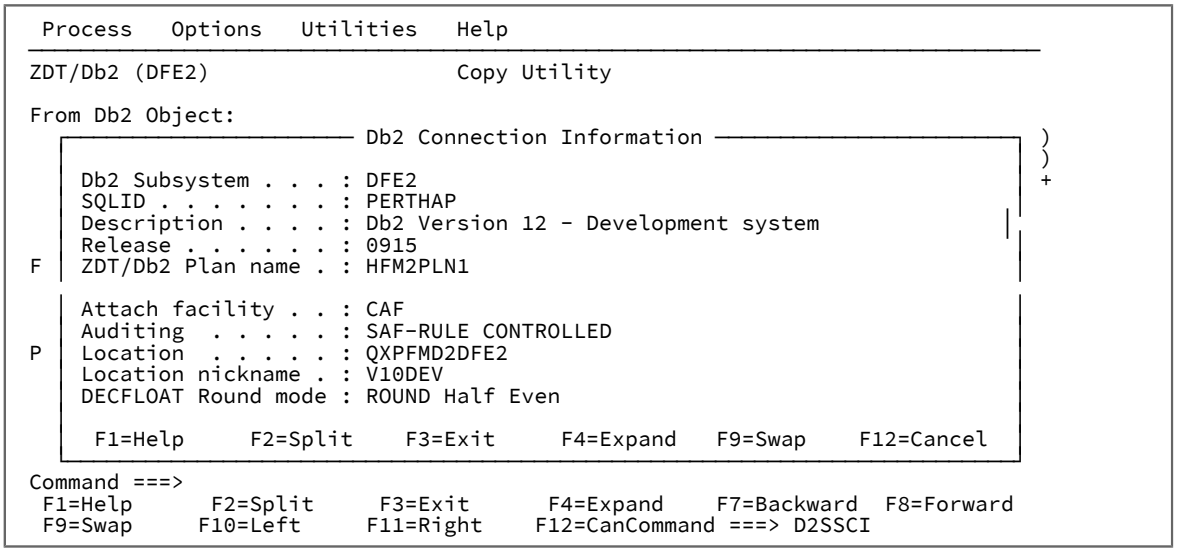
When you enter a value in this field, synonym object names are not recognized and are not included in the selection list.

It is also possible to specify the location of a Db2® system using:

- The Db2® subsystem ID
- A location nickname defined for the Db2® system

Note: In what follows, the use of either the Db2® subsystem ID, or a location nickname, is referred to as an *alternative location reference*.

In order to be able to use these alternatives, the systems programmer must have defined the location value and location nickname in the HFM2SSDM macro definition for each Db2® system (HFM2POPT module). You can check whether a location and location nickname have been defined by selecting **About Db2** from the Help pull-down menu.



If a location value has been specified, you can use the Db2® subsystem ID (for example, DFE2) as an alternative to the actual location name. If location and location nickname values have been specified, you can use the location nickname (for example, V9DEV) as an alternative to the actual location name.

You cannot use wildcards when specifying an alternative location reference. When you specify an alternative location reference, the actual location appears in any location reference displayed by ZDT/Db2; that is, in the Db2® object name shown in a Z Data Tools/Db2 editor session. Similarly, any Db2® object name specified in a Db2® batch job is the actual location name. Specifying an alternative location reference in a Z Data Tools/Db2 batch job is not supported.

You can only access remote locations that have been defined to your local Db2® subsystem.

❷ **Owner**

The authorization ID of the owner of the table, view, or alias.

If you leave this field blank, ZDT/Db2 uses the current SQL ID. Otherwise, type the full name of the owner, an asterisk (*) to match any owner, or a pattern using wild card characters.

To specify a synonym in the object name field, either leave this field blank, or specify a generic name that includes the current SQLID, or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the table space, database, or location fields are non-blank.

Note: Db2® alias names are recognized. Db2® synonyms are not recognized.

❸ **Name**

The name of the Db2® object.

This entry field is mandatory. Type the full name of the Db2® object, an asterisk (*) to display the names of all available objects, or a pattern using wild card characters.

To specify a synonym in this field, type the name of the synonym and ensure that the owner field is either blank, a generic name that includes the current SQLID, or the current SQLID. Synonyms are not recognized or included in the table selection list if any of the table space, database, or location fields are non-blank.

❹ **Database**

The name of the database to which you want to restrict your selection.

Leave this field blank if you do not want to confine the selection to specific databases. Otherwise, type the full name of the database, an asterisk (*) to match any database, or a pattern using wild card characters.

When you enter a value in this field, synonym names are not recognized or included in the selection list.

❺ **Table space**

The name of the table space to which you want to restrict your selection.

Leave this field blank if you do not want to confine the selection to specific table spaces. Otherwise, type the full name of the table space, an asterisk (*) to match any table space, or a pattern using wild card characters.

When you enter a value in this field, synonym names are not recognized or included in the selection list.

If you specify an asterisk (*) or a pattern in one or more of the preceding entry fields (**Owner**, **Name**, **Database**, **Table space**), ZDT/Db2 displays a selection list showing the Db2® tables and views that match *all* the selection criteria entered in these fields. The selection list may also include synonyms entries under circumstances discussed above.

To select the table, view, or synonym you want, type `S` in the selection column next to the required entry.

For example, say you specified `__DB2` in the **Owner** entry field, `*EMP*` in the **Name** entry field, and `FM*` in the **Database** entry field. ZDT/Db2 displays a selection list of Db2® tables and views where:

- The name of the owner is five characters long, with *"Db2®"* as the last three characters,
- The name of the Db2® table or view contains the characters "EMP", and
- The name of the database starts with "FM".

To select, say, the view VEMP, type `S` in the selection column next to that entry.

Related tasks

Using an asterisk (*) or a pattern in entry fields on page 50

Related references

Table/View/Alias Selection panel on page 890
Db2 Location Selection panel on page 621

Using an asterisk (*) or a pattern in entry fields

For entry fields on ZDT/Db2 panels where you can enter an asterisk (*) or wildcards, you can use:

- An asterisk (*) on its own to match all values, or
- A pattern using wildcard characters to select all objects that match the pattern.

In either case, ZDT/Db2 displays a selection list. To select the listed item you want, type `S` against the required entry and press Enter.

You can specify a pattern using the following characters:

asterisk (*) in combination with one or more other wildcard characters

Represents a string of zero or more characters. Acts the same as a percent sign (%).

percent sign (%)

Represents a string of zero or more characters. Acts the same as an asterisk (*).

underscore (_)

Represents any single character, including a trailing blank.

Other characters

Any other characters that are valid for the entry field.

The following examples show how various patterns work if used to select entries from the following list:

- AGE
- AGREE
- GUM
- IGLOO
- MAN
- MANE
- MANY
- MANAGE
- MANAGES

Pattern

Entries selected

AGE

AGE, MANAGE, and MANAGES

MAN_

MAN, MANE, and MANY

MANAGE%

MANAGE and MANAGES

_G*

AGE, AGREE, and IGLOO

%G__

AGE, MANAGE, and MANAGES

Specifying the SQLID

You can specify the SQLID to be used as part of the connection to Db2® by typing the required value in the **SQL ID** field on the ZDT/Db2 Primary Option Menu panel. The default is your TSO logonid.

You cannot change the SQLID value until ZDT/Db2 is successfully connected to a Db2® system.

You require authorization to change the SQLID value to something other than your TSO logonid. See your Db2® systems administrator if you require this capability.

You can also change the SQLID value dynamically by using the SQLID primary command.

You can determine the current value for the SQLID by selecting "About Db2®" from the Help pull-down menu.

Related references

Primary Option Menu panel on page 760
SQLID primary command on page 1006

Specifying a data set and a member name

Some ZDT/Db2 panels require you to specify the data set that you wish to use for the current process. If the data set is a PDS, a CA-Panvalet library, or a library accessed using the Library Management System Exit, you also must enter the member name. These panels are collectively referred to as Entry panels.

To specify your data set on any entry panel:

1. In the **Data set name** field, enter a fully-qualified data set name or a name pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated Member field must be empty.

   Note: In the case of a copybook, this can be the name of a CA-Panvalet library or the name of a library accessed using the Library Management System Exit.

   If you do not enclose the data set name in quotation marks, then the TSO prefix for your user ID is used as the high-level qualifier.

   If you enter a pattern, when the Entry panel is processed ZDT/Db2 displays a list of the data sets that match the pattern.

Patterns can contain a percent sign (%) to represent a single character, and an asterisk (*) to represent any number of characters within one qualifier. Two asterisks (**) represent any number of characters within any number of qualifiers.

For performance reasons, you should qualify the data set name as much as you can.

2. If you specified the name of a partitioned data set (PDS), the name of a CA-Panvalet library, or the name of a library accessed using the Library Management System Exit, in the **Data set name** field (without including a member name or name pattern in parenthesis), then enter a name or name pattern in the **Member** field.

   A member name pattern can consist of any characters that are valid in a member name and the following two special pattern characters:

   asterisk (*)

   Represents any number of characters. As many asterisks as required can appear anywhere in a member name. For example, if you enter `*d*`, a list of all members in the data set whose name contains "d" is displayed.

   percent sign (%)

   A place-holding character representing a single character. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter `%%%%`, a list of all members in the data set whose name is four characters in length is displayed.

   If you perform either of these actions:
   - Do not include a member name or name pattern in parenthesis in the **Data set name field**, and leave the **Member** field blank, or you specify a pattern in the **Member** field,
   - Include a member name pattern in parenthesis in the **Data set name field**, and leave the **Member** field blank,

   ZDT/Db2 displays a member name list when the Entry panel is processed and the data set has been specified or selected.
3. For data sets that have not been catalogued, enter the serial number of the volume which contains the data set in the **Volume serial** field.
4. Complete the remaining panel-specific fields, and then press Enter to process the panel function.

   If both the data set name and the member name have been fully specified, the panel action will proceed to its next stage.

   If you entered a pattern for the data set name or member name, the Data Set Selection panel is displayed, followed by the Member Selection panel, as needed. When you have made your selections from these panels, the Entry panel action will proceed to its next stage.

5. When the Data Set Selection panel is displayed, select your data set by entering an `s` in the **Sel** field adjacent to the required data set. You can only select one name from this list.
6. When the Member Selection panel is displayed, select your members by typing an `s` in the **Sel** field next to each member, and then pressing Enter.

You can select as many names as required from this list. When the panel is processed, the first member on the list is used in the relevant function. When the function is completed, the next member in list is used. For example, if you are selecting several members to be edited, the first member is displayed in the Edit Session panel. When you exit this panel, the next member is displayed, and so on, until all selected members have been processed.

Examples of data set name patterns

The examples below assume that the following data sets exist under the USERID high-level qualifier:

```
USERID.HFM.DATA
USERID.HFM.TEMPLATES
USERID.COBOL.COPY
USERID.COBOL.SOURCE
USERID.PLI.COPY
USERID.PLI.SOURCE
USERID.MISC.DATA.BACKUP
USERID.WORK
```

Example 1

Entering the following pattern (with a single asterisk):

```
'USERID.*'
```

displays the only item with exactly one level of qualifier after USERID:

```
USERID.WORK
```

Example 2

Entering the following pattern (with two asterisks):

```
'USERID.**'
```

displays the complete list of data sets with the USERID high-level qualifier.

Example 3

Entering the following pattern:

```
'USERID.*.%%%%'
```

displays the following list (containing those data sets with exactly three qualifiers, whose third-level qualifier contains exactly four characters):

```
USERID.HFM.DATA
USERID.COBOL.COPY
USERID.PLI.COPY
```

Examples of specifying a member name or pattern

The examples below assume that the following members exist within the data set named, HFMUSER.DATA:

```
DATA1
HFMCDATA
HFMCTAM
```

```
HFMCTEM
HFMCTMP
NEWDATA
NEWSTUFF
TEMPA
TEMPB
```

Example 1

Entering the following details on an Entry panel:

```
⋮
   Data set name . . . . . HFMUSER.DATA
   Member  . . . . . . . . ________
⋮
```

displays the entire member name list:

```
DATA1
HFMCDATA
HFMCTAM
HFMCTEM
HFMCTMP
NEWDATA
NEWSTUFF
TEMPA
TEMPB
```

Example 2

Entering the following details on an Entry panel:

```
⋮
   Data set name . . . . . HFMUSER.DATA
   Member  . . . . . . . . HFMCT*
⋮
```

displays the member name list:

```
HFMCTAM
HFMCTEM
HFMCTMP
```

Example 3

Entering the following details on an Entry panel:

```
⋮
   Data set name . . . . . HFMUSER.DATA(NEWDATA)
   Member  . . . . . . . . _________
⋮
```

displays details for the member NEWDATA in data set HFMUSER.DATA.

Example 4

Entering the following details on an Entry panel:

```
⋮
   Data set name . . . . . HFMUSER.DATA(TEM*)
   Member  . . . . . . . . _________
⋮
```

displays the member name list:

```
TEMPA
TEMPB
```

Example 5

Entering the following details on an Entry panel:

```
⋮
   Data set name . . . . . HFMUSER.DATA(*)
   Member  . . . . . . . . _________
⋮
```

displays the entire member name list:

```
DATA1
HFMCDATA
HFMCTAM
HFMCTEM
HFMCTMP
NEWDATA
NEWSTUFF
TEMPA
TEMPB
```

Related references

Copy Utility ("From") panel on page 513
Copy Utility ("To") panel on page 521
Data Create Utility panel on page 603
Db2 View panel on page 631
Db2 Browse panel on page 612
Db2 Edit panel on page 616
Export Utility ("To") panel on page 701
Export Utility ("From") panel on page 698
Import Utility ("To") panel on page 720
Import Utility ("From") panel on page 719
LOAD from panel on page 731
LOAD Utility - Using Templates panel on page 735
UNLOAD Utility (Tables) "To" panel on page 911
Print Utility panel on page 764

Canceling long-running Db2® queries

On occasions, when you are working with very large amounts of Db2® data, you may experience long response times associated with the ZDT/Db2 query that ZDT/Db2 is processing.

When you are using one of the following ZDT/Db2 utilities, you can use the *attention interrupt key* to cancel the query.

- View
- Browse
- Edit
- Print
- Copy
- Import
- Export
- Create
- SQL

Pressing the attention interrupt key when Db2® is processing cancels the current Db2® function. Any uncommitted changes made since the start of the unit of work are lost.

Getting help

ZDT/Db2 offers two levels of help information: Tutorial Help and, for input fields (and some output fields), field-level help information. Tutorial Help is a context-sensitive help system that provides detailed explanations of the ZDT/Db2 panels and the processes in which they are used. The field-level help information provides a description of the current field. In addition, all error messages have an associated full text explanation, which can help you to determine the cause of a problem.

This section explains how to find the help information that you need.

Related tasks

Accessing Tutorial Help on page 57
Displaying field-level help information on page 60
Displaying the expanded form of error messages on page 60

Accessing Tutorial Help

The Tutorial Help system is both context-sensitive and structurally organised. You can get directly to the information you need, using the context-sensitive access method or you can enter the Help system at a specific location and navigate to any topic information within the structure.

To access a context-sensitive Tutorial Help page:

1. Place your cursor on the Command Line or anywhere in the panel that is outside of a field entry line.
2. Press the Help function key (F1). The Tutorial Help page associated with the current panel is displayed.

   When you access the Tutorial Help in this way, the first page that is displayed may be a main topic page, with a menu listing the associated sub-topics, or it may be a sub-topic page, depending upon the context from which you started.

3. Navigate through the Tutorial Help pages, to find the information you require (see below for a list of navigation commands).
4. When you have finished, press the Exit function key (F3) to return to your starting panel.

The Tutorial Help system is organised with a Table of Contents structure that is based upon the ZDT/Db2 Primary Options menu. Each entry within the Table of Contents leads to a main topic with a number of associated sub-topics. In addition, there is an index that lists selected topics from the Tutorial Help, alphabetically by subject.

To choose your starting point in Tutorial Help:

1. Select the Help pull-down menu from a panel Action Bar.
2. Enter the option number for the Tutorial Help entry point that you require. These are:

1. Help for help

Displays the Help panel for the Tutorial Help system.

2. Extended help

Displays the Tutorial Help panel associated with the current ZDT/Db2 panel (equivalent to pressing F1 from the ZDT/Db2 panel). When on the Primary Options menu, this is the Tutorial Help Table of Contents panel.

3. Keys help

Displays a panel that provides help for the Function Keys that are active on the current ZDT/Db2 panel.

4. Help index A-M

Displays the Help Index for subjects starting with "A" to "M".

5. Help index N-Z

Displays the Help Index for subjects starting with "N" to "Z".

6. Tutorial

Displays the Tutorial Help Table of Contents panel.

7. About

Displays the ZDT/Db2 version, PTF level, and copyright information in a window.

8. About Db2®

Displays the Db2® subsystem, Db2® version and release information, ZDT/Db2 plan names, and audit options in a window.



Note: For more information about audit options, see SAF-rule controlled auditing on page 71.

9. News about ZDT/Db2

Displays a panels providing general information about the current ZDT/Db2 version/release.

To navigate Tutorial Help, enter one of the following commands in the Command line on any Tutorial page:

BACK or B

To back up to the previously viewed page.

SKIP or S

To skip the current topic and go on to the next topic.

UP or U

To display a higher level list of topics.

TOC or T

To display the Table of Contents.

INDEX or I

To display the Tutorial Index. Once the Index has been displayed, use the Right function key (F11) and the Left function key (F10) to scroll through the list, then position your cursor on a subject and press Enter to display the topic.

Alternatively, you may use the following keys whenever you are in the tutorial:

Enter (Ctrl)

To display the next sequential page within a topic.

Help (F1)

To display the Help Tutorial page within the Tutorial Help system (that is, the page that describes how to use Tutorial Help).

End (F3)

To terminate the tutorial.

Up (F7)

To display a higher level list of topics (instead of typing UP).

Down (F8)

To skip to the next topic (instead of typing SKIP).

Right (F11)

To display the next sequential page within a topic (instead of pressing Enter).

Left (F10)

To display the previous sequential page within a topic (instead of typing BACK).



Note: The listed keys are the default key mappings. As you can customize key mappings, they may be different on your system.

Related references

Primary Option Menu panel on page 760

Displaying field-level help information

To display the field-level help information for a field on a Z Data Tools/Db2 panel:

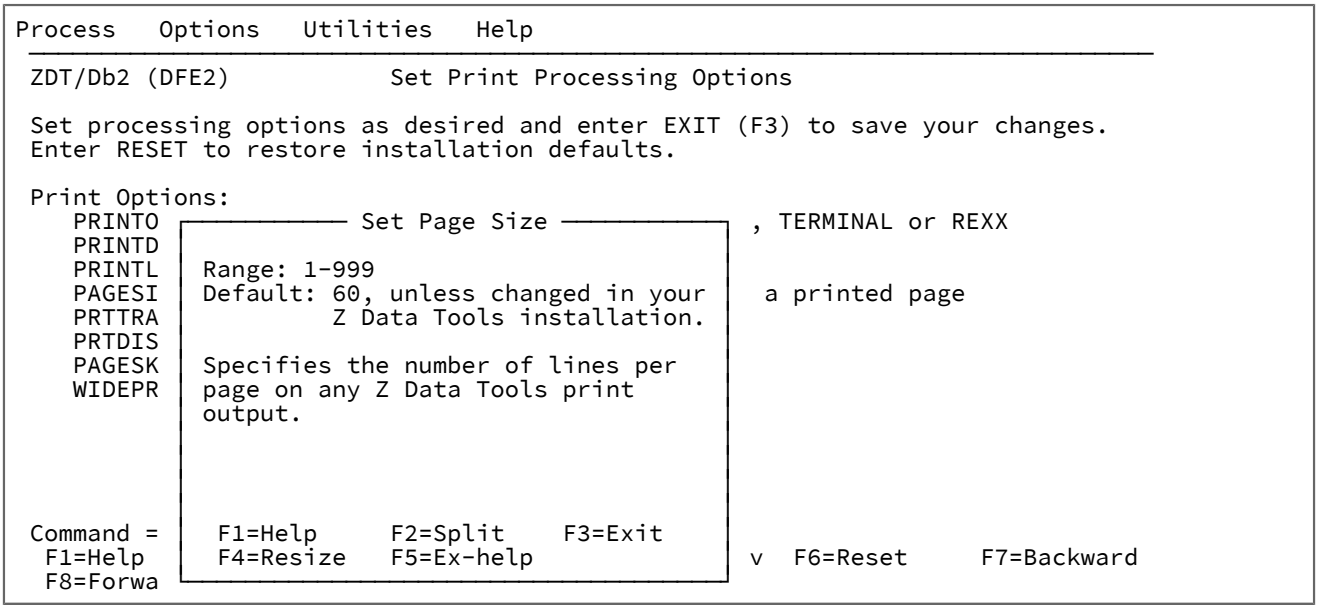
1. Place your cursor on the field entry line.
2. Press the Help function key (F1). The field-level help information is displayed in a pop-up window.

   If further information is needed, press the Extended Help function key (F5) to access the Tutorial Help page for the current panel.

   Pressing Help while the field information window is displayed takes you to the Help Tutorial page within the Tutorial Help system (that is, the page that describes how to use Tutorial Help).
3. When you have finished reading the message or the relevant Tutorial Help page, press the Exit function key (F3) to return to your starting panel.

For example, Figure 16: Field-level help information for the PAGESIZE field on page 60 shows the message seen when F1 is pressed from the PAGESIZE field in the Set Print Processing Options panel.

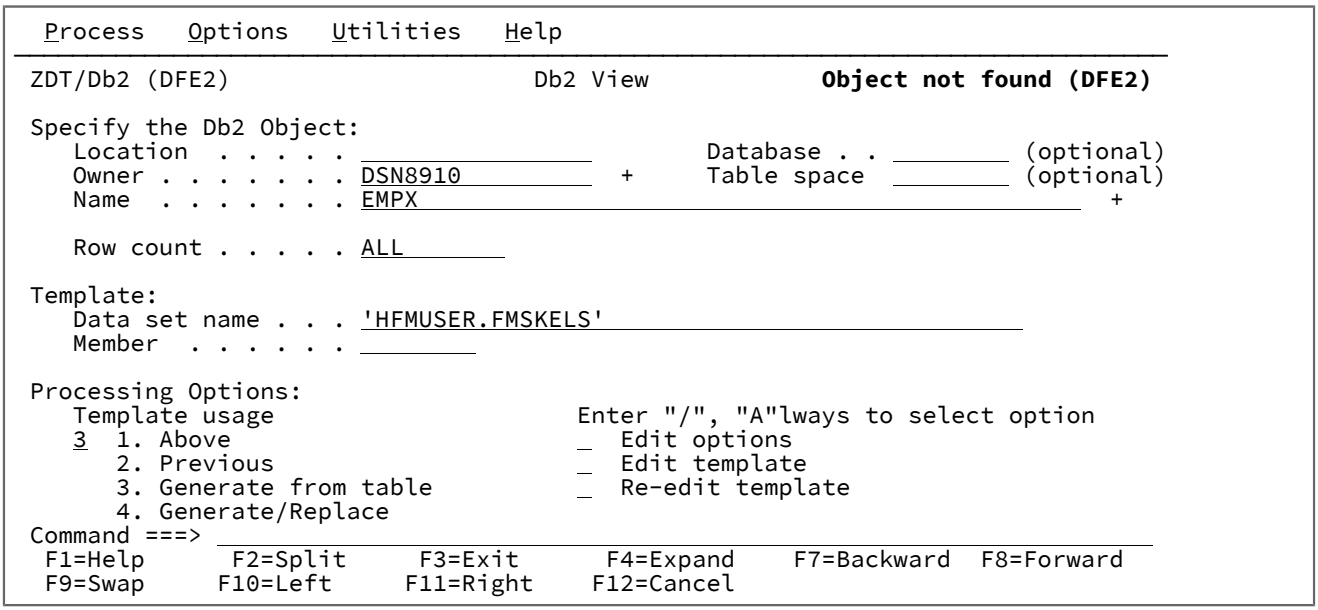
Figure 16. Field-level help information for the PAGESIZE field



Displaying the expanded form of error messages

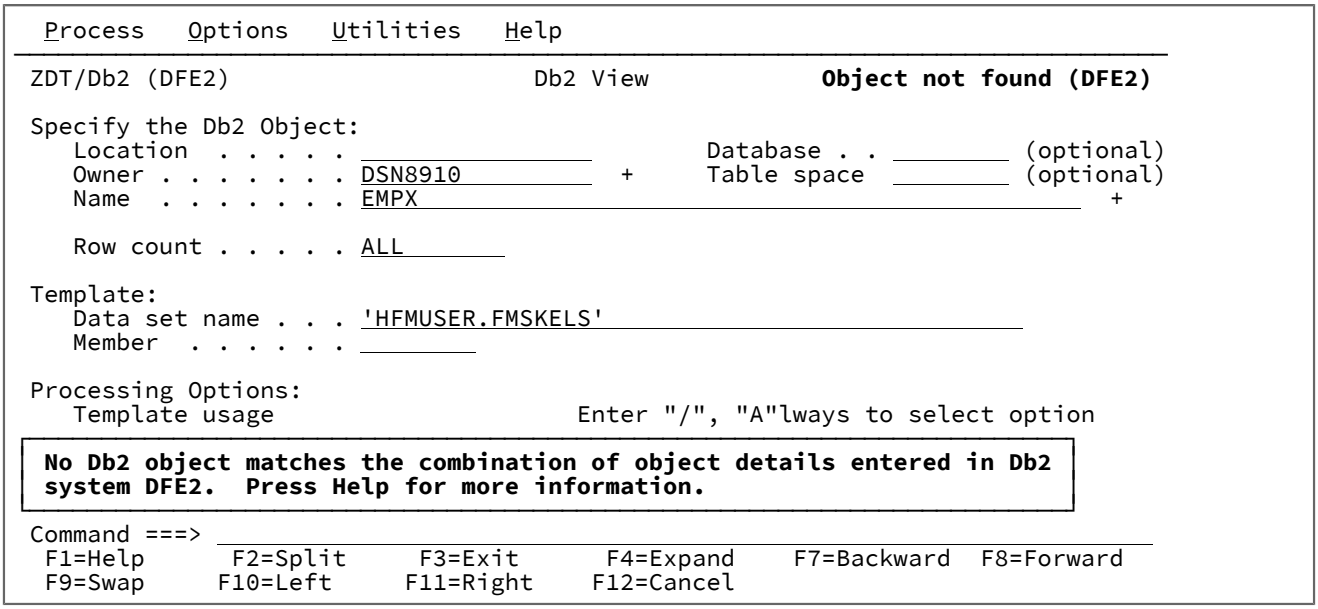
If an error occurs when ZDT/Db2 attempts to process a panel, a short text message displays in the upper right corner of the screen. Figure 17: Db2 View showing short text message on page 61 shows an example where ZDT/Db2 cannot find the Db2® object specified.

Figure 17. Db2® View showing short text message



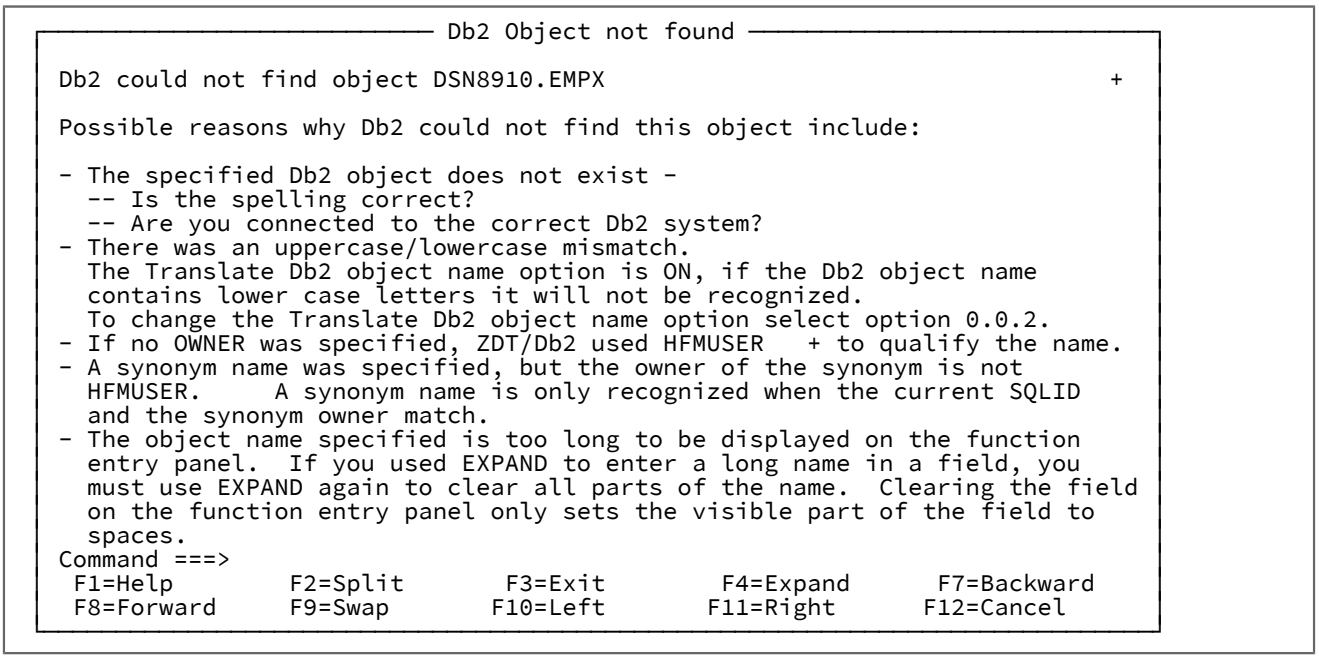
While this message is displayed, pressing the Help function key (F1) displays the expanded text of the error message at the bottom of the screen as shown in Figure 18: Db2 View showing expanded message on page 61.

Figure 18. Db2® View showing expanded message



Some messages, like the one shown in Figure 18: Db2 View showing expanded message on page 61 have "Press Help for more information." in the text of the long message. In this case, you can press the Help function key (F1) (again) when the long message is displayed to show a help panel with additional information like that shown in Figure 19: Help panel displaying additional information relating to a long error message on page 62.

Figure 19. Help panel displaying additional information relating to a long error message



Setting default processing options

Many of the processing operations performed by Z Data Tools use default values that can be set from within the ZDT/Db2 application. By adjusting these values, you can customize ZDT/Db2 so that its behavior is best suited to your needs.

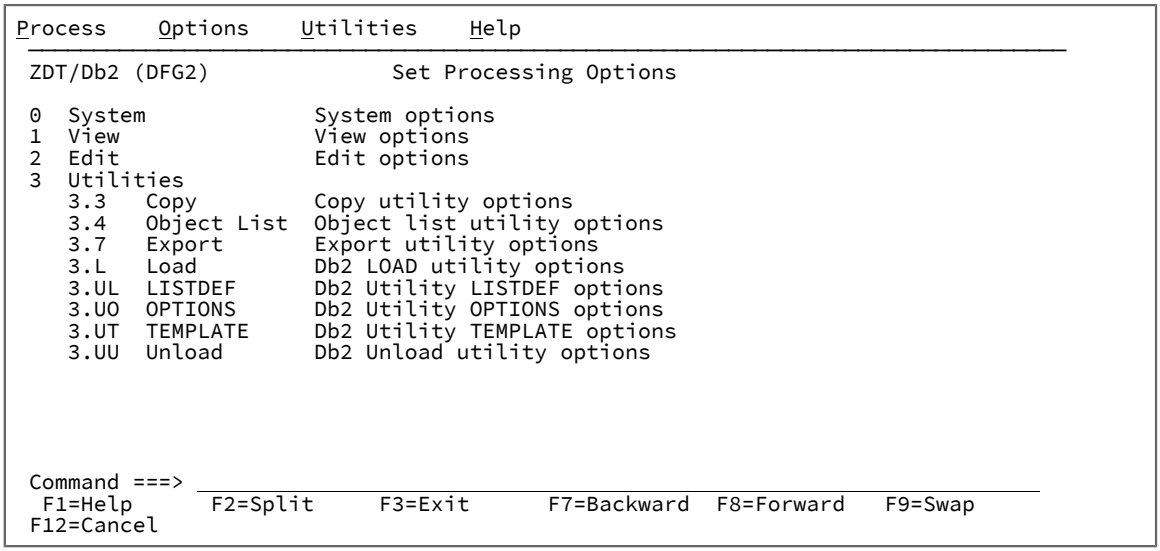
When you first use ZDT/Db2, you will probably find that the default values for processing options (as set at the time of installation) are sufficient. As you become more familiar with working with ZDT/Db2, you may choose to change some of these options to better suit your needs.

You can update these default values by accessing the relevant processing option panel (for example, the ZDT/Db2 System Options panel, Editor Options panels, and so on). Changes are saved in your ISPF user profile for future ZDT/Db2 sessions. They remain in effect until you change the option again.

To access a processing option panel, use either of the following methods:

- From the ZDT/Db2 Primary Option Menu panel:
  1. Select Option 0 (Settings). The Set Processing Options panel is displayed.

     Figure 20. Set Processing Options menu



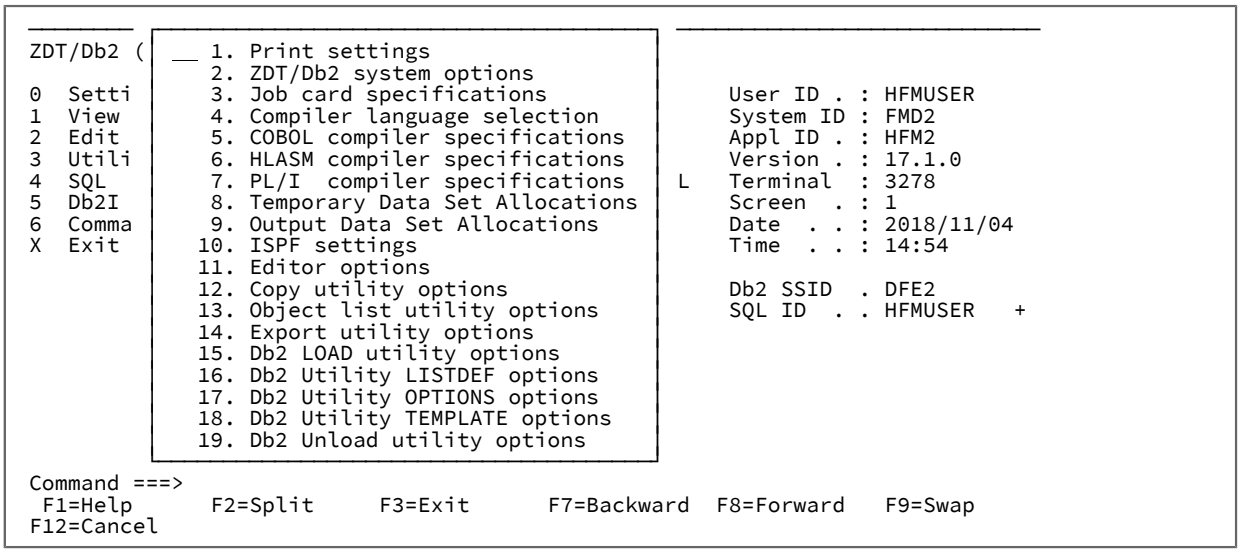
  2. Select the required processing options type from the menu.

     For example, to display the current ZDT/Db2 system options, enter `0.2` on the command line and press Enter to display the ZDT/Db2 System Options (1 of 3) panel.

     Note: You can navigate directly to the options panel you want by entering the path a shown on the Set Processing Options panel (such as `0.2`, `3.7`, and so on). Alternatively, for the system and utility options, you can enter `0` or `3` to first display the Set System Processing Options or Set Utility Processing Options panels respectively from which can then select the option you want.

- From any ZDT/Db2 panel, use the **Options** pull-down menu to select the required processing options type:

  Figure 21. Options pull-down menu



When you select an option processing type, Z Data Tools displays a panel listing those options and their current values. The following notes apply to all of these option panels:

- To change the value of an option, overtype its current value.
- To save the options you have changed and exit the panel, press the Exit function key (F3).
- To exit a panel without saving changes, press the Cancel function key (F12).
- The options on these panels only affect the behavior of ZDT/Db2 panels.

Related tasks

Selecting an item from the action bar on page 40

Related references

Set Processing Options panel on page 832
Set System Processing Options panel on page 835
Set Utility Options panel on page 840

System Options (option 0)

System options displays the Set System Processing Options panel from which you can select and display subsequent system options panels.

Related references

Set System Processing Options panel on page 835
Set Print Processing Options panel on page 827
Systems Options (1 of 4) panel on page 857
Set Batch Job Card Information panel on page 817
Compiler Language Selection panel on page 506
Set COBOL Processing Options panel on page 818
Set PL/I Processing Options panel on page 825
Set Temporary Data Set Allocation Options panel on page 837
Set Output Data Set Allocation Options panel on page 824
Set Trace options panel on page 839
ISPF Settings panel on page 727

Print settings (option 0.1)

The Print settings options control such things as the destination of your print output and the page format to use.

Related tasks

Printing the contents of a Db2 table on page 328

Related references

Set Print Processing Options panel on page 827

ZDT/Db2 system options (option 0.2)

The ZDT/Db2 system options control such things as the translation of Db2® object names and input SQL statements to uppercase, the use of double quotation marks for SQL identifiers, and the external format for TIME columns.

Related references

Systems Options (1 of 4) panel on page 857
Systems Options (2 of 4) panel on page 859
Systems Options (3 of 4) panel on page 862

Job card specifications (option 0.3)

On the Set Batch Job Card Information panel, you can specify batch job card information to be used for generating batch job submission JCL. The Batch Job Card Information panel provides a number of blank lines (**Batch Submission Job Statement Information**) where you can enter this information.

ZDT/Db2 assumes any non-blank lines define a JCL job card and copies the information from these lines into generated JCL. If all the lines are blank, the JCL generation process generates a basic job card using information from system variables.

Related references

Set Batch Job Card Information panel on page 817

Compiler language selection (option 0.4)

The Compiler language selection options allow you to select which compiler is used to generate templates from the COBOL or PL/I copybook for ZDT/Db2 import and export utilities.

Related tasks

How ZDT/Db2 compiles a copybook into a template on page 80
Copying data from a VSAM or QSAM file on page 251
Copying data to a VSAM or QSAM file on page 279

Related references

Compiler Language Selection panel on page 506

COBOL compiler specifications (option 0.5)

The COBOL compiler specifications control various options used by the COBOL compiler to generate templates from COBOL copybooks used by the ZDT/Db2 import and export utilities.

Related tasks

- How ZDT/Db2 compiles a copybook into a template on page 80
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279

Related references

- Set COBOL Processing Options panel on page 818

HLASM compiler specifications (option 0.6)

The HLASM compiler specifications control various options used by the HLASM compiler to generate templates from HLASM copybooks used by the ZDT/Db2 import and export utilities.

Related tasks

- How ZDT/Db2 compiles a copybook into a template on page 80
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279

Related references

- Set HLASM Processing Options panel on page 821

PL/I compiler specifications (option 0.7)

The PL/I compiler specifications control various options used by the PL/I compiler to generate templates from PL/I copybooks used by the ZDT/Db2 import and export utilities.

Related tasks

- How ZDT/Db2 compiles a copybook into a template on page 80
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279

Related references

- Set PL/I Processing Options panel on page 825

Temporary Data Set Allocations (option 0.8)

The temporary data set allocations allow you to specify UNIT, DATA CLASS, STORAGE CLASS, and MANAGEMENT CLASS names for the allocation of a temporary data set.

Related references

Set Temporary Data Set Allocation Options panel on page 837

Output Data Set Allocations (option 0.9)

The output data set allocations allow you to specify UNIT, DATA CLASS, STORAGE CLASS, and MANAGEMENT CLASS names for the allocation of an output data set.

Related references

Set Output Data Set Allocation Options panel on page 824

Trace options (option 0.10)

The Trace options displays the Set Trace options panel.

Related references

Set Trace options panel on page 839

ISPF settings (option 0.11)

The ISPF settings option displays the ISPF Settings panel.

Related references

ISPF Settings panel on page 727

View and Edit options (options 1 and 2)

The View and Edit options (both these options lead you to the first of the Editor Options panels) control various options used to format the display of data when viewing or editing.

Note: If you are viewing or editing data, and you use the **Options** pull-down menu to change these options, then your changes do not take effect until your next ZDT/Db2 editor session. To temporarily change the behavior for the current ZDT/Db2 editor session only, use the **Edit options** option on the Db2® View or Db2® Edit panels.

Related tasks

Viewing and changing Db2 data on page 122
Setting options for the current ZDT/Db2 editor session on page 126

Related references

Editor Options (1 of 8) panel on page 644

Utilities (option 3)

Utilities displays the Set Utility Processing Options panel from which you can select and display subsequent utility options panels.

Related references

- Set Utility Options panel on page 840
- Copy Options panel on page 509
- Export Options (1 of 3) panel on page 689
- LOAD Utility Options panel on page 737
- LISTDEF Options panel on page 729
- OPTIONS Options panel on page 750
- TEMPLATE Options (1 of 2) panel on page 897
- UNLOAD Options panel on page 905

Print utility options (option 3.1)

The Print utility options control such things as print mode (table or single, null column indicator character, batch execution, uncommitted read, and hexadecimal representation.

Related tasks

- Printing the contents of a Db2 table on page 328

Related references

- Print Utility Options panel on page 763

Copy utility options (option 3.3)

The Copy utility options control such things as concurrency, table locking, and duplicate key processing when copying data between Db2® tables.

Related tasks

- Copying data from one Db2 object to another on page 246

Related references

- Copy Options panel on page 509

Object List Options (option 3.4)

The object list utility options affect the behavior of catalog table displays produced in the following utility functions:

- Object List utility (3.4)
- Manage Db2® Privileges (3.5)
- Explain Utilities (4.5)

Related references

Object List Options (1 of 2) panel on page 743
Object List Options (2 of 2) panel on page 746

Export utility options (option 3.7)

The Export utility options allow you to tailor the way in which the Export utility operates.

Related tasks

Copying data to a VSAM or QSAM file on page 279

Related references

Export Options (1 of 3) panel on page 689
Export Options (2 of 3) panel on page 692
Export Options (3 of 3) panel on page 696

Db2® LOAD utility options (option 3.L)

The Load utility options control the way in which ZDT/Db2 generates JCL for the Db2® LOAD utility.

Related tasks

LOAD utility on page 357

Related references

LOAD Utility Options panel on page 737

Db2® Utility LISTDEF options (option 3.UL)

The Listdef utility options displays the LISTDEF Options panel.

Related tasks

Setting options for Db2 utilities on page 348

Related references

LISTDEF Options panel on page 729

Db2® Utility OPTIONS options (option 3.UO)

The Option utility options displays the OPTIONS Options panel.

Related tasks

Setting options for Db2 utilities on page 348

Related references

OPTIONS Options panel on page 750

Db2® Utility TEMPLATE options (option 3.UT)

The Template utility options displays the TEMPLATE Options panel.

Related tasks

Setting options for Db2 utilities on page 348

Related references

TEMPLATE Options (1 of 2) panel on page 897
TEMPLATE Options (2 of 2) panel on page 898

Db2® Unload utility options (option 3.UU)

The Unload utility options displays the Unload Options panel.

The Unload Utility options apply when you use the LOAD utility in Utilities (3.9) to transfer data.

Related tasks

Setting options for Db2 utilities on page 348
UNLOAD utility on page 380

Related references

UNLOAD Options panel on page 905

Displaying or editing DBCS characters

If you want to display or edit DBCS characters in ZDT/Db2 and you have a DBCS capable terminal, you must ensure that your ISPF settings specify a terminal type that supports these characters.

To specify the Terminal Type in ISPF:

1. Exit from ZDT/Db2 and return to the z/OS® Primary Options Menu panel.

   Although you can change the ISPF Terminal Type setting while ZDT/Db2 is running, the change will not be reflected until the next time that ZDT/Db2 is invoked.

2. Select option `0. Settings`. The ISPF Settings panel is displayed.
   Figure 22. ISPF Settings panel

```
  Log/List  Function keys  Colors  Environ  Workstation  Identifier  Help
──────────────────────────────────────────────────────────────────────────────
                              ISPF Settings
                                                              More:   -
  _  Session Manager mode                  Command delimiter . ;
  /  Jump from leader dots
  _  Edit PRINTDS Command
  /  Always show split line
  _  Enable EURO sign

Terminal Characteristics
  Screen format   1  1. Data    2. Std     3. Max     4. Part

  Terminal Type   3    1. 3277       2. 3277A      3. 3278       4. 3278A
                       5. 3290A      6. 3278T      7. 3278CF     8. 3277KN
                       9. 3278KN    10. 3278AR    11. 3278CY    12. 3278HN
                      13. 3278HO    14. 3278IS    15. 3278L2    16. BE163
                      17. BE190     18. 3278TH    19. 3278CU    20. DEU78
                      21. DEU78A    22. DEU90A    23. SW116     24. SW131
                      25. SW500

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=Actions  F12=Cancel
```

3. Select either 3277KN or 3278KN from the list of Terminal Types. (You may need to scroll down this panel to find the list of Terminal Types.)
4. Press Exit (F3) to save your selection and return to the z/OS® Primary Options Menu panel.
5. Restart ZDT/Db2.

Related tasks

Starting ZDT/Db2 on page 26
Exiting from ZDT/Db2 on page 29

SAF-rule controlled auditing

The way in which auditing is performed during a Z Data Tools/Db2 session may be determined by SAF rules, depending on the auditing requirement of your installation.

Determining when SAF-rule controlled auditing is in effect

You can determine if SAF-rule controlled auditing is in effect by selecting "**About Db2®**" on the Help pull-down menu:

- When SAF-rule controlled auditing is effect, the **Auditing** line of the "**About Db2®**" pop-up panel shows "SAF-RULE CONTROLLED".
- When SAF-rule controlled auditing is active and no auditing is to occur, the message indicates that auditing is prohibited.
- When SAF-rule controlled auditing is active and auditing might occur, the message states only that SAF-rule controlled auditing is active.

Note: The final decision as to whether to audit or not depends on the ZDT/Db2 function and the appropriate SAF XFACILIT rules defined for the Db2® object names, SQL statements, or Db2® commands that are accessed by the ZDT/Db2 function.

SAF rules for auditing

When auditing is determined by SAF rules:

- Auditing can be specified independently for each Db2® system accessed by ZDT/Db2, to either your audit log data set, to SMF, or to both destinations.
- All ZDT/Db2 functions that allow the execution of SQL statements are subject to audit, with the exception of SQL statements used to access the Db2® catalog tables for the internal processes of ZDT/Db2.
- SAF rules control if and where the **Create audit trail (*)** option is displayed on ZDT/Db2 panels. "(*)" is appended to the **Create audit trail** option to show auditing is under SAF control.

For select statements applying to a Db2® object, SAF rules can be specified at a Db2® object name level allowing audit for READ, UPDATE, or both types of operations. An UPDATE operation includes any SQL statement that changes Db2® data such as INSERT, DELETE, and UPDATE.

SAF-rule controlled auditing not in effect

When SAF-rule controlled auditing is *not* in effect (that is, auditing is controlled by the settings in HFM2POPT), the **Create audit trail** option is displayed on the first *global* editor options panel. The value shown reflects the current behaviour. When you display the first *local* editor options panel, the **Create audit trail** option is not displayed since it appears on the editor function entry panel.

SAF-rule controlled auditing in effect, no auditing

When SAF-rule controlled auditing *is* in effect, and *no* auditing is to occur for the Db2® system, the **Create audit trail (*)** option is not displayed on either the editor function entry panel, or on the first editor options panel.

SAF-rule controlled auditing in effect, auditing possible

When SAF-rule controlled auditing *is* in effect, and auditing *is* to occur for the Db2® system, the actual decision on whether or not to create an audit log depends on the Db2® object name being processed.

Therefore, when you press Enter on the editor function entry panel, a check is made to see if auditing should occur for the current editor type (browse or edit) and the Db2® object name.

If the **Create audit trail (*)** option is set appropriately (that is, it matches the auditing requirement as specified in the relevant SAF rule), processing continues, the data is displayed, and audit records are written, or not, depending on the option selection.

If you have not selected the **Create audit trail (*)** option and auditing is required, the function entry panel is re-displayed with a message to indicate that auditing is required. You will not be able to process the object until the audit option is selected.

If you have selected the **Create audit trail (*)** option and auditing is not required, the function entry panel is re-displayed with a message to indicate that auditing is not required. You will not be able to process the object until the audit option is un-selected.

Note that both the previous scenarios assume that you do not have audit control access for the Db2® object in question. If you do have control level audit access for the Db2® object, the current value of the **Create audit trail (*)** option determines whether or not audit records are written.

Related references

Editor Options (1 of 8) panel on page 644
Db2 Edit panel on page 616
Db2 View panel on page 631

Chapter 4. Working with templates

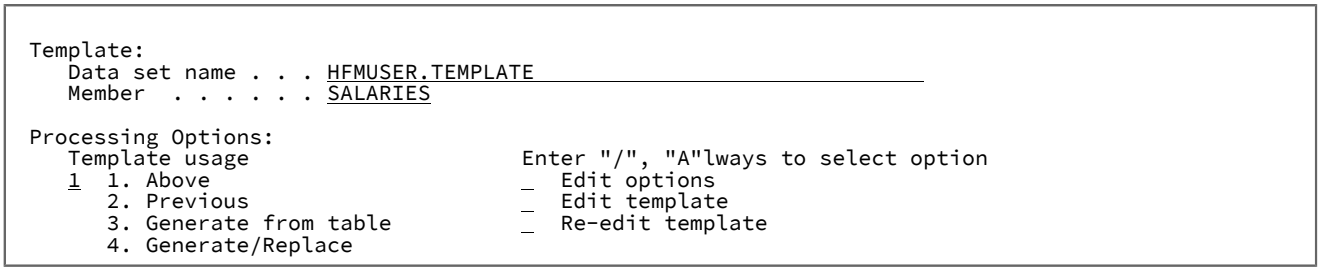
ZDT/Db2 uses *templates* to provide a formatted view of your data, enabling you to view, edit, and manipulate data according to the columns and data types in the table you are working with.

Initially, ZDT/Db2 generates a template from the Db2® catalog table. After you have saved the template, you can then use ZDT/Db2 to enhance it.

The following sections describe how you use templates with ZDT/Db2.

Associating a template with a Db2® object

To use an existing template with a Z Data Tools/Db2 panel, you supply the name of the template in the template **Data set name** entry field (and optionally the template **Member** entry field) then, in the **Template usage** entry field, select option 1 (Above).



If you have previously used a template with the input Db2® object, you can select option 2 (Previous) instead. The Previous option instructs ZDT/Db2 to ignore the contents of the Template fields, and instead use the template most recently associated with the input data set. If you want to edit the template before using it, select the **Edit template** field.

To generate a template for the specified Db2® object, select option 3 (Generate from table). This option instructs ZDT/Db2 to ignore the contents of the Template fields and use information from the Db2® catalog table to generate a template for the Db2® object. This is the default setting.

To generate a template for the specified Db2® object and immediately save the template, enter the name you want ZDT/Db2 to use for the template in the template **Data set name** entry field (and optionally the template **Member** entry field) then, in the **Template usage** entry field, select option 4 (Generate/Replace).

Related tasks

Editing a template on page 84

Related references

Copy Utility ("From") panel on page 513
Copy Utility ("To") panel on page 521
Data Create Utility panel on page 603
Db2 Browse panel on page 612
Db2 Edit panel on page 616

Db2 View panel on page 631
Export Utility ("To") panel on page 701
Export Utility ("From") panel on page 698
Import Utility ("To") panel on page 720
Import Utility ("From") panel on page 719
LOAD from panel on page 731
Print Utility panel on page 764
UNLOAD Utility (Tables) "From" panel on page 910
UNLOAD Utility (Tables) "To" panel on page 911

Information stored in a template

The following section describes the information that a template can contain, and where and how ZDT/Db2 uses the information.



Note:
In this chapter, *"transferring data"* refers to using any of the following ZDT/Db2 functions:

- Copy (option 3.3)
- Import (option 3.6)
- Export (option 3.7)
- Utilities (option 3.9), Db2® LOAD utility, Db2® UNLOAD utility

Row selection criteria

Determines which rows are to be selected.

Rows that do not match the row selection criteria are excluded when comparing, copying[2] or printing.

Column selection

Determines which columns are displayed (if you are browsing or editing) or printed.

Except for when you are exporting data, this information is ignored when transferring or creating Db2® objects. (When you are copying, importing, exporting, or loading data, if you want to exclude some data in the input from being transferred to the output, you need to *"map"* the *"To"* template without those fields.)

Column sequence

2. When copying, ZDT/Db2 uses this information in the “From” template. In a “To” template, this information is ignored. However, when comparing, the selection information in both the “Old” and “New” templates is used.

Overrides the default order in which columns are displayed (if you are browsing or editing) or printed. By default, columns are displayed or printed in the order in which the columns are defined in the object in the Db2® catalog (the COLNO column of SYSIBM.SYSCOLUMNS).

This information is ignored when transferring or creating Db2® objects. (When copying, importing, exporting, or loading data, if you want to exclude some data in the input from being transferred to the output, you need to *"map"* the *"To"* template without those fields.)

Column headings

Overrides the default column headings when browsing, editing or printing. (The default column headings are the column names defined in the Db2® catalog.)

Column fixed attributes

The column name and data type as defined in the Db2® catalog on which the template was based.

Column use attributes

Consist of two attributes:

Output width

Defines the number of positions allocated to the column if you are browsing, editing or printing.

The default width depends on the data type and relates to the *"external"* format for the data. For example, the default width for a SMALLINT column is 6 bytes (the usual external display size).

If a numeric field contains a valid numeric value, but the value is too large to fit in the output width, then, if you are browsing or editing, the value is truncated and highlighted.

Leading zeros (numeric columns only)

Defines whether or not the column value is shown with leading zeros if you are browsing, editing or printing.

Column create attributes

Defines the value to which the column is initialized if you are transferring data, or creating a table.

If you are transferring data, ZDT/Db2 uses this information in the *"To"* template. In a *"From"* template, this information is ignored.

Column mapping

Defines which columns in the *"From"* template map to columns in the *"To"* template if you are transferring data.

If you are transferring data, ZDT/Db2 uses this information in the *"To"* template. In a *"From"* template, this information is ignored.

Scrambling options

Determine how the contents of the column are scrambled (if at all) during a copy, import, or export process.

Related tasks

- Copying data from one Db2 object to another on page 246
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279
- LOAD utility on page 357
- UNLOAD utility on page 380
- Mapping data on page 112

Related references

- Db2 data types on page 440
- Copy Utility ("From") panel on page 513
- Import Utility ("From") panel on page 719
- Export Utility ("From") panel on page 698
- Data Create Utility panel on page 603
- Db2 Utilities panel on page 627

Printing a template

To print a template, use the TP primary command.

Related tasks

- Looking at the print output from your ZDT/Db2 session on page 333

Related references

- TP primary command on page 1007
- PB primary command on page 986

Using templates with non-Db2 data

Some ZDT/Db2 functions interface with non-Db2 data:

- You can *retrieve* non-Db2 data with these ZDT/Db2 functions:
  - Import (option 3.6) — for retrieving non-Db2 data from a VSAM or QSAM data set.
  - The Db2® LOAD utility in Utilities (option 3.9) — for retrieving data from a sequential data set.
- You can *write* data to non-Db2 files with these ZDT/Db2 functions:
  - Export (option 3.7) — for writing data to a VSAM data set, a sequential data set, or a member of a partitioned data set.
  - The Db2® UNLOAD utility in Utilities (option 3.9) — for unloading data to a sequential data set.

Because this is non-Db2 data, ZDT/Db2 is not able to create a template from the Db2® catalog table. Instead, you have to provide a source for the template. The source is either a (non-Db2) template or a copybook. This describes the records and fields in the data set or member. If you specify a copybook, ZDT/Db2 compiles it into a template. You can save it and reuse it with any application data set that has the same record structure.

A copybook is a data set containing COBOL, HLASM, or PL/I source statements that describe the record structure of an application data set:

COBOL data description entries

Each level-01 group item in the copybook describes a record type in the application data set; elementary items in the group describe fields in the record type.

For example, the following entries describe the record structure of an application data set that contains two record types, ORDER and ITEM:

```
01 ORDERS.
   02 ORDER-ID        PIC X(5).
   02 CUSTOMER-ID     PIC X(5).
   02 ORDER-DATE.
      03 ORDER-YEAR   PIC X(4).
      03 ORDER-MONTH  PIC X(2).
      03 ORDER-DAY    PIC X(2).
01 ITEM.
   02 PRODUCT-ID      PIC X(9).
   02 QUANTITY        PIC 9(4) BINARY.
   02 UNIT-COST       PIC 9(8) BINARY.
```

ORDER records contain five fields: ORDER-ID, CUSTOMER-ID, ORDER-YEAR, ORDER-MONTH and ORDER-DAY. ITEM records contain three fields: PRODUCT-ID, QUANTITY and UNIT-COST.

HLASM DSECT definitions

HLASM copybooks are similar to COBOL copybooks, in that the major structures (DSECT names) describe record types and elementary names describe fields:

```
ORDER      DESCT
ORDERID    DS  CL9
CUSTOMERID DS  CL5
ORDERDATE  DS  0CL8
ORDERYEAR  DS  CL4
ORDERMONTH DS  CL2
ORDERDAY   DS  CL2
ITEM       DESCT
PRODUCTID  DS  CL9
QUANTITY   DS  H
UNITCOST   DS  F
```

PL/I DECLARE statements

Similar to COBOL copybooks, major structures (level-1 names) describe record types, and elementary names describe fields:

```
DECLARE 1 ORDER,
          2 ORDER_ID       CHAR(9),
          2 CUSTOMER_ID    CHAR(5),
          2 ORDER_DATE,
            3 ORDER_YEAR   CHAR(4),
            3 ORDER_MONTH  CHAR(2),
            3 ORDER_DAY    CHAR(2);
DECLARE 1 ITEM,
          2 PRODUCT_ID     CHAR(9),
          2 QUANTITY       BIN(15),
          2 UNIT_COST      BIN(31);
```

A copybook can contain the entire source of a COBOL, HLASM, or PL/I program, or only COBOL data description entries, HLASM DSECT definitions, or PL/I DECLARE statements. ZDT/Db2 ignores everything in the copybook except for COBOL data description entries, HLASM DSECT definitions, or PL/I DECLARE statements.

ZDT/Db2 supports copybooks stored in any of the following:

- PDS
- PDSE
- CA-Panvalet library
- Library accessed using the Library Management System Exit

Note:

1. Library members may not be packed by ISPF.
2. For details about enabling Z Data Tools to use a CA-Panvalet library, or a library accessed using the Library Management System Exit, see the *Z Data Tools Customization Guide*.

Related tasks

How ZDT/Db2 compiles a copybook into a template on page 80

How ZDT/Db2 compiles a copybook into a template

If you specify a copybook, ZDT/Db2 performs the following steps:

1. Checks (if the compiler language selected is not HLASM) whether the copybook is a complete source program, or contains only COBOL data declarations or PL/I DECLARE statements. If the copybook is not a complete program, ZDT/Db2 includes the copybook in a shell program.
2. Calls the selected compiler to check the syntax of the copybook, and to produce an ADATA file.

   You can use the Settings: Compiler language selection panel (option 0.0.4) to specify whether ZDT/Db2 calls the COBOL, HLASM, or PL/I compiler.

   For COBOL and PL/I copybooks, you can select the **Auto detect** option on the Compiler Language Selection panel to instruct Z Data Tools to detect the language of the copybook and call the appropriate compiler. In this case, ZDT/Db2 analyzes the source to determine whether the language of the source is COBOL or PL/I and creates a template based on the result.

   To instruct Z Data Tools to process an Assembler copybook, you must set the language option on the Compiler Language Selection panel to HLASM. Once you have set the option to HLASM, every function that supports copybooks can be used with an Assembler copybook. Use the HLASM compiler specifications to provide extra SYSLIB and change ALIGN and DBCS processing options.

   You can specify the acceptable return code level for the compilation in the compiler specifications panel for the selected language.

   If the compilation produces a return code with a value less than or equal to the specified maximum, ZDT/Db2 processes the information in the ADATA file to create a template.

   If the compilation produces a return code with a value greater than the specified maximum, ZDT/Db2 displays a pop-up menu. From the pop-up menu, you can:
   - View the compilation listing using Print Browse (option 3.11).
   - Cancel the template creation process.

- Try the compilation again. Before you select this option, first view the compilation listing and correct any errors in the copybook. While you are viewing the compilation listing, you can use the ISPF split screen facility to swap to another ISPF session and use your editor to correct the errors in the copybook.
- Ignore the errors and proceed with creating the template. This option is only available if ZDT/Db2 is able to create a template. Some compilation errors, such as warnings, might have no effect on the creation of the template. If you are unsure, view the compilation listing. If the compilation error is too severe, ZDT/Db2 cannot create a template.
- Try the compiler for the other language. The option is not available if the HLASM compiler language has been selected.

Related tasks

Compiler language selection (option 0.4) on page 65
Compiler Language Selection panel on page 506

Related references

Set COBOL Processing Options panel on page 818
Set PL/I Processing Options panel on page 825
Set HLASM Processing Options panel on page 821

Support for COBOL data description entries

In the COBOL data description entries specified in a copybook, ZDT/Db2 supports:

- ASSIGN clause
- OCCURS clause (including OCCURS DEPENDING ON)
- PICTURE clause (except for numeric editing)
- REDEFINES clause
- RENAMES clause
- USAGE clause
- VALUE clause

ZDT/Db2 ignores:

- BLANK WHEN ZERO clause
- DATE FORMAT clause
- EXTERNAL clause
- GLOBAL clause
- JUSTIFIED clause
- Numeric editing in the PICTURE clause
- SYNCHRONIZED clause

ZDT/Db2 supports all data item level numbers except 88 (condition-names).

Support for HLASM data description entries

In the HLASM data description entries specified in a copybook, ZDT/Db2 produces a record, or segment, for each DSECT it finds with corresponding DS/DC fields.

For more information, see the section "Using HLASM copybooks with Z Data Tools" in the Base *Z Data Tools User's Guide and Reference*.

Support for PL/I data description entries

In the PL/I data description entries specified in a copybook, ZDT/Db2 supports:

- PL/I data attributes VARYING and VARYINGZ for single and double-byte character strings.
- PL/I bit variables including the VARYING attribute.
- Binary, decimal, floating point and numeric picture data (excluding symbols).
- REFER variables for array extents. Both upper and lower bound array elements may be declared for each dimension. The number of extents for each dimension is always:

  ```
  upper bound - lower bound + 1
  ```

  Note: A zero extent is not allowed for a PL/I dimension. If a bound is not specified PL/I assumes a value of 1.

- REFER variable to declare the length of a string variable, character (AN), varying-length character (VC) or zero-terminated character string (ZC) in bytes.
- ZDT/Db2 also supports PL/I double-byte character variables (DB), varying-length double-byte character (VD) and zero-terminated double-byte characters (Z2) as byte pairs. ZDT/Db2 also supports bit strings (BT) and varying bit string (VB) variables.
- REFER variables can be declared as integer, fixed binary, packed decimal, floating point, or character string variables.
- UNION declarations at any level which can have varying length (REFER string length or REFER array extents).

  Note: UNIONS can contain overlays of different data types, which may require template workbench record selection criteria or SHOW control if data is not displayable.

ZDT/Db2 ignores:

- INIT data values
- REFER start expression, for example,

```
VAR(, expr Refer abc, )  or
CHAR(expr Refer abc).
```

- EXTERNAL attributes
- DEFINED structure overlays
- LIKE structure attributes

Note: In general, ZDT/Db2 treats Numeric Edited fields as alphanumeric. A special code, ZA, is used for unsupported zoned decimals.

Support for variable-length arrays

A variable-length array is defined in a COBOL copybook by an OCCURS DEPENDING ON (ODO) clause, and in a PL/I copybook by a dimension attribute where the lower or upper bounds (or both) are defined by REFER options.

The following considerations apply if you are using a template that contains variable-length arrays:

- A record structure can contain multiple variable-length arrays; however, the fields that define the size of the arrays (the ODO or refer *"objects"*) must all be in the record before the first variable-length array item.
- If any of the following are true:
  - (COBOL copybook only.) The object is out of the range specified by the ODO clause.
  - The record length does not match the calculated length of a record with the number of array items specified by the object.

  then, if you are editing the data set, the prefix area contains the description `=LGTH`, indicating that the record was not selected because of incorrect length.
- If you change the value of an object field, then the number of items in the array expands or contracts automatically.

  New items are inserted at the end of the current array (and before any fields that follow the array); numeric items are initialized to zero, and alphanumeric items are initialized to blanks.

  Similarly, items are deleted from the end of the array, and any fields subsequent to the end of the table are not affected.

  For PL/I, the number of items in an array is determined by subtracting the lower bound of the dimension attribute from the upper bound and then adding one. If both the lower and upper bounds are specified using REFER options, changing the value of either of the refer object fields inserts or deletes array items accordingly.
- (COBOL copybook only.) You cannot change the object field to a value outside the range specified by the ODO clause. (ZDT/Db2 supports an ODO lower value of 0.)
- (COBOL copybook only.) Inserting a new record results in a record with the minimum number of array items specified by the ODO definition. For example:
  - Given the following ODO clause:

    ```
    ODO-ONE OCCURS 0 TO 10 TIMES DEPENDING ON ODO-CNTRL-ONE.
    ```

if you insert a new record, the ODO-CNTRL-ONE field is initialized to zero, with no ODO-ONE array items in the new record.

- Given the following ODO clause:

```
ODO-ONE OCCURS 5 TO 7 TIMES DEPENDING ON ODO-CNTRL-ONE.
```

ODO-CNTRL-ONE is initialized to 5, with five ODO-ONE array items in the new record.

- (PL/I copybook only.) Inserting a new record results in a record with zero array items. Upper and lower bound refer object fields are initialized to zero.

Related tasks

Copying data from a VSAM or QSAM file on page 251
LOAD utility on page 357
Copying data to a VSAM or QSAM file on page 279

Editing a template

ZDT/Db2 always uses a template to display Db2® data.

The discussion in this section provides examples based on editing a template while you are browsing the sample data shown in Figure 142: Sample data on page 867.

You can use either of the following methods to edit a template:

- On an entry panel where you can use templates:
  1. Specify the template you want ZDT/Db2 to use (by selecting the appropriate **Template usage** option and, if required, specifying the name of the template).
  2. Select the **Edit template** field.
  3. Press Enter.

  ZDT/Db2 presents the template for editing, then runs the function after you have finished editing.

  If you edit a template in this way, you can change all parts of the template (including row selection criteria).
- If you are using Browse (option B), View (option 1), or Edit (option 2) to browse, view, or edit data, you can use the TEDIT primary command.

Note:

1. You can *only* use the TEDIT primary command from within a Z Data Tools/Db2 editor session (if you have the data displayed). You cannot use TEDIT from either the Db2® View or Db2® Edit function panels.
2. If you use the TEDIT primary command, you cannot change the **Order**, **A/D**, or **Row Selection Criteria** input fields.

The reason you cannot alter these parts of the template is that a Z Data Tools/Db2 editor session is in progress and the session can contain uncommitted changes to data. Altering the preceding parts of the template might require data to be retrieved from Db2® using a new (and possibly different) SQL statement.

When you edit a template (using either of the preceding methods), ZDT/Db2 displays the Column Selection/Edit panel.

When you have finished editing the template using the Column Selection/Edit panel, you can:

- Use the edited template *"on the fly"* without saving it by pressing the RunTemp function key (F6),
- Optionally save the edited template before returning to the ZDT/Db2 function panel by pressing the Exit function key (F3), or
- Return to the ZDT/Db2 function panel without saving the template by pressing the Cancel function key (F12).

Related tasks

Where you can use templates on page 22
Viewing and changing Db2 data on page 122

Related references

TEDIT primary command on page 1006
Column Selection/Edit panel on page 497

Adding a description to a template

When you are creating or modifying a template, you can issue the DESCRIBE primary command to add a description to the template.

To add or modify a description for a template:

1. Display the Column Selection/Edit panel for the template concerned by:
   - Selecting the **Edit template** field on an entry panel where you can the template, or
   - Issuing the TEDIT primary command from within a Z Data Tools/Db2 editor session that is using the template to display the data
2. Enter the DESCRIBE primary command.

   ZDT/Db2 displays the Db2® Template Description panel.

3. Enter or modify the description for the template in the **Description** entry field.
4. Press the Exit function key (F3) to return to the Column Selection/Edit panel.

Related references

DESCRIBE primary command on page 938
TEDIT primary command on page 1006
Column Selection/Edit panel on page 497
Db2 Template Description panel on page 626

Selecting rows

Before you view or edit data, you can select the rows you want to view by editing a template and specifying *row selection criteria*.

To specify row selection criteria, you must first display the Column Selection/Edit panel by selecting the **Edit template** field on an entry panel where you can use templates. (If you use the TEDIT primary command to display the Column Selection/Edit panel, you cannot change the row selection criteria.)

You can specify row selection criteria:

- By column, or
- In freeform style

The following sections describe these two methods.

For a full-screen edit, use the SQL primary command or press the SQL function key (F4).

Row selection by column

To enter row selection criteria by column, type `1` on the command line on the Column Selection/Edit panel and press Enter. ZDT/Db2 displays the Row Selection Criteria panel.

On the line for each column for which you want to specify selection criteria, you specify an operator (like = or >) in the **Op** column and, in most cases, a value in the **Value** column.

For example, to select only employees with a surname of SMITH from a table of employees:

1. On the line for the column LASTNAME, type `=` in the Op column and `SMITH` (optionally enclosed in single quotation marks) in the Value column:
   Figure 23. Example 1: Specifying a simple predicate to match a single value

```
Cmd Con ( Column name                    Data type(length)    Op  Value
    <-> - <---+----1----+----2----+----> <---+----1----+----> <-> <---+----
*** **** Top of data  ****
         EMPNO                           CHARACTER(6)
    AND  FIRSTNME                        VARCHAR(12)
    AND  MIDINIT                         CHARACTER(1)
    AND  LASTNAME                        VARCHAR(15)          =   SMITH
```

This predicate works well in the situation where the data in the table is stored in uppercase. However, if the data in the table is in mixed case (for example, 'Smith') or contains errors (for example, 'SmiTH'), Db2® will not return all the desired rows.

Normally, any operator that is entered in the Op column has no effect on the column name. The following two operators allow a column function to be specified:

LKL

Results in this expression: LOWER(colname) LIKE value.

LKU

Results in this expression: UPPER(colname) LIKE value.

In the following example, any upper case character in the value will result in no matches, so ZDT/Db2 automatically converts the value entered into lower case. For more information, see The expression generated from the expression in example 2 on page 88.

Figure 24. Example 2: Specifying the LKL operator

```
Cmd Con ( Column name                    Data type(length)    Op  Value
     <-> - <---+----1----+----2----+----> <---+----1----+----> <-> <---+----1----+----2
 *** ****  Top of data  ****
           EMPNO                           CHARACTER(6)
     AND   FIRSTNME                        VARCHAR(12)
     AND   MIDINIT                         CHARACTER(1)
     AND   LASTNAME                        VARCHAR(15)          LKL '%SMITH%'
```

In the following example, any lower case character in the value will result in no matches, so ZDT/Db2 automatically converts the value entered into upper case. For more information, see The expression generated from the expression in example 3 on page 88.

Figure 25. Example 3: Specifying the LKU operator

```
Cmd Con ( Column name                    Data type(length)    Op  Value
    <-> - <---+----1----+----2----+----> <---+----1----+----> <-> <---+----
*** ****  Top of data  ****
          EMPNO                           CHARACTER(6)
    AND   FIRSTNME                        VARCHAR(12)
    AND   MIDINIT                         CHARACTER(1)
    AND   LASTNAME                        VARCHAR(15)          LKU '%smith%'
```

You can also specify a column expression directly if required. To enter a column expression, you can either issue the EDITCOL editor primary command, or type the EC prefix command against the column name where an expression is required. After issuing these commands, the column name field changes to an input field.

Figure 26: Example 4: Entering a column expression and a data value with the escape character on page 88 shows the use of a column expression and the '\' escape character. This example demonstrates how to display the rows where an employee was hired in March. The escape character is used to prevent ZDT/Db2 from attempting to validate the data in the value field according to the type. The column function MONTH returns an integer representing the month (1 - 12), whereas the data type for the column is DATE.

Figure 26. Example 4: Entering a column expression and a data value with the escape character

```
Cmd Con ( Column name                    Data type(length)    Op  Value
    <-> - <---+----1----+----2----+----> <---+----1----+----> <-> <---+----
*** ****  Top of data  ****
          EMPNO                          CHARACTER(6)
    AND   FIRSTNME                       VARCHAR(12)
    AND   MIDINIT                        CHARACTER(1)
    AND   LASTNAME                       VARCHAR(15)
    AND   WORKDEPT                       CHARACTER(3)
    AND   PHONENO                        CHARACTER(4)
    AND   MONTH(HIREDATE)                                     =   \3
```

2. Press the Exit function key (F3).

ZDT/Db2 displays the Column Selection/Edit panel showing the generated row selection criteria (WHERE clause) in the **Row Selection Criteria** field at the top of the panel:

The results of the expressions entered in example 1 - example 4 are shown below:

Figure 27. The expression generated from the expression in example 1.

```
 Process   Options   Utilities   Help
-------------------------------------------------------
ZDT/Db2 (DFG2)                                        Co
Command ===>
TABLE DSN81010.EMP
--------------------------------- Row Selection Criteria
1 Sel: WHERE "LASTNAME" = 'SMITH'
```

Figure 28. The expression generated from the expression in example 2

```
 Process   Options   Utilities   Help
-------------------------------------------------------
ZDT/Db2 (DFG2)                                        Co
Command ===>
TABLE DSN81010.EMP
--------------------------------- Row Selection Criteria
1 Sel: WHERE LOWER("LASTNAME") LIKE '%smith%'
```

Figure 29. The expression generated from the expression in example 3

```
 Process   Options   Utilities   Help
-------------------------------------------------------
ZDT/Db2 (DFG2)                                        Co
Command ===>
TABLE DSN81010.EMP
--------------------------------- Row Selection Criteria
1 Sel: WHERE UPPER("LASTNAME") LIKE '%SMITH%'
```

Figure 30. The expression generated from the expression in example 4

```
 Process   Options   Utilities   Help
-------------------------------------------------------
ZDT/Db2 (DFG2)                                        Co
Command ===>
TABLE DSN81010.EMP
--------------------------------- Row Selection Criteria
1 Sel: WHERE MONTH(HIREDATE) = 3
```

3. Press Exit function key (F3)

ZDT/Db2 displays the selected data:

Figure 31. The data resulting from the expressions in example 1 - example 4

```
ZDT/Db2 (DFG2)                                              Table Edit
Command ===>
2 rows - End of object.
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PUN-+> <-N-+----1-> -       <-N-+----1----> <-N      <-->    <---+---->
****** ****  Top of data  ****
000001 000250 DANIEL<      S       SMITH<          D21      0961    1969-10-30
000002 000300 PHILIP<      X       SMITH<          E11      2095    1972-06-19
```

Figure 32. The data resulting from the expression in example 4

```
ZDT/Db2 (DFG2)                                              Table Edit
Command ===>
3 rows - End of object.
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PUN-+> <-N-+----1-> -       <-N-+----1----> <-N      <-->    <---+---->
****** ****  Top of data  ****
000001 000200 DAVID<               BROWN<          D11      4501    1966-03-03
000002 000280 ETHEL<       R       SCHNEIDER<      E11      8997    1967-03-24
000003 200280 EILEEN<      R       SCHWARTZ<       E11      8997    1967-03-24
```

Continuing with the expression entered in Figure 23: Example 1: Specifying a simple predicate to match a single value on page 86, if you now want to select, for example, employees with a surname of SMITH or JONES, you need to modify the row selection criteria. To do this, you must return to the Row Selection Criteria panel:

1. From the panel of displayed data, press the Exit function key (F3) to return to the Db2® Edit entry panel.
2. Select the **Edit template** field and press the Exit function key (F3) to display the Column Selection/Edit panel.

   You will notice on this panel that the **Row Selection Criteria** field (showing your previous WHERE "LASTNAME" = 'SMITH' clause) is protected. This is because row selection criteria entered by column or as a free-form expression are not interchangeable. You can change row selection criteria that has been entered by column, by returning to the Row Selection Criteria panel. You can convert row selection criteria entered by column to a free-form expression by issuing the SQL command on the Column Selection/Edit panel, and saving the data. The conversion removes the specification by column. You cannot convert a free-form expression to a row selection criteria entered by column.
3. Enter `1` on the command line to display the Row Selection Criteria panel.
4. To select more than one value for the same column (as in this case, where you want to select both SMITH and JONES), there are two methods you can use:
   - Use multiple lines for the column:
     a. Type the R (Repeat) prefix command in the **Cmd** column to repeat the line for the column name concerned
     b. On the repeated line (or lines) for the column name, change the connector (in the **Con** column) to *"OR"* by overtyping the AND with `O` or `OR`
     c. Type the value in the **Value** column

   For our example, the selection lines now look like this:

```
Cmd Con ( Column name                          Op  Value                        )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
⋮
___ AND _ LASTNAME                             =   SMITH                        _
___ OR  _ LASTNAME                             =   JONES                        _
```

- Use the IN operator:
  a. On the line for LASTNAME, change the operator (from =) to IN. The IN operator lets you specify a list of values you want to select.
  b. In the value column, type the values you want to select. For alphanumeric columns, when you specify a list of values, each value *must* be enclosed in quotation marks and separated by a comma.

  For our example, the selection line now look like this:

```
Cmd Con ( Column name                          Op  Value                        )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
⋮
___ AND _ LASTNAME                             IN  'SMITH','JONES'              _
```

5. Press the Exit function key (F3) twice to display the selected data. Whichever method you used to specify row selection criteria for SMITH or JONES, the same results are achieved:

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Table Edit                    Template saved
TABLE HFMUSER.EMP                                                  Format TABL
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+---->
000000 ****  Top of data  ****
000001 000210 WILLIAM<     T       JONES<          D11      0942    11.04.1979
000002 000250 DANIEL<      S       SMITH<          D21      0961    30.10.1969
000003 000300 PHILIP<      X       SMITH<          E11      2095    19.06.1972
000004 ****  End of data  ****
⋮
```

Continuing with this same example, if you now want to select employees with a surname of SMITH or JONES who are in department D21, you again need to modify the row selection criteria.

This time, in order to avoid any ambiguity, you may need to use parentheses to define the sequence of evaluation for the resultant WHERE clause. On the Row Selection Criteria panel, you can use the **(** and **)** columns to enclose groups of expressions in parentheses.

For our example, the selection lines may now look like this:

```
Cmd Con ( Column name                          Op  Value                        )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
⋮
___ AND ( LASTNAME                             =   SMITH                        _
___ OR  _ LASTNAME                             =   JONES                        )
___ AND _ WORKDEPT                             =   D21                          _
```

The selected data now appears as:

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                    Table Edit                    Template saved
TABLE HFMUSER.EMP                                                 Format TABL
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+---->
000000 ****  Top of data  ****
000002 000250 DANIEL<      S       SMITH<          D21      0961    30.10.1969
000004 ****  End of data  ****
⋮
```

Changing the display (80-character screens only)

On wide screens, ZDT/Db2 displays both the **Column Name** and the **Data type(length)** details for each column on the Row Selection Criteria panel. However, for 80-character screens, by default ZDT/Db2 displays only the **Column Name** details.

On an 80-character screen, there are five different versions of the Row Selection Criteria panel:

1. The default screen, which includes the column name and no data type field.
2. A screen that includes the data type field, but not the column name field.
3. A screen that includes the column name field and an abbreviated data type field.
4. A screen that includes a truncated column name field and no data type field.
5. A screen that is similar to version 4. However, the column name data is right-aligned, rather than left-aligned.

Use the PF10 and PF11 function keys to cycle between the five screen versions that are described in the preceding list. Pressing PF10 from screen 1 displays screen 5. Pressing PF11 from screen 5 displays screen 1.

Note: The cursor must not be positioned on a **Value** input field.

Verifying the format of DATE, TIME, and TIMESTAMP columns

If the row selection criteria you are specifying includes a DATE, TIME, or TIMESTAMP column, the value you specify (in the **Value** entry field) for that column must be in the correct format.

If you are not sure of the format you should use:

1. Type the required operator (such as "=") in the **Op** entry field
2. Type "=" in the **Value** entry field
3. Press Enter

ZDT/Db2 displays the field in the format required, showing the current value.

For example, if you enter "=" in the **Value** entry field for a TIMESTAMP(4) column, ZDT/Db2 returns the current date and time in the format required for a TIMESTAMP(4) column:

```
Cmd Con ( Column name                          Op  Value                        )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
⋮
___ AND — TSTAMP                               =   2004-09-24-09.19.36.7749
⋮
```

Note that the last two digits are removed. TIMESTAMP(4) means four digits of precision.

Using the same format, you can now overtype the value that ZDT/Db2 returns with the value you want to use as part of your row selection criteria.

Related tasks

Scrolling the Row Selection Criteria field on page 94
Expanding the Row Selection Criteria field on page 95

Related references

Column Selection/Edit panel on page 497
Row Selection Criteria panel on page 799

Row selection criteria in freeform style

To enter freeform row selection criteria, you use the **Row Selection Criteria** entry field at the top of the Column Selection/Edit panel. The entry field is protected and you cannot update it if:

- It already contains a WHERE clause which was generated by column, or
- You have used the TEDIT command to display the Column Selection/Edit panel

In this entry field, enter any valid SQL WHERE clause to specify selection criteria to select the data you want to view or edit.

For example, if you have two columns, WORKDEPT and LASTNAME, and you want to display only details of people in department A00 whose family name starts in the range *"A"* to *"M"*:

1. Enter the following selection criteria:

   ```
   WORKDEPT = 'A00' AND LASTNAME BETWEEN 'A%' AND 'M%'
   ```

   Alternatively, you can use the ZDT/Db2 reference numbers as abbreviations for the column names. For example, you can write the preceding expression as:

   ```
   #5 = 'A00' AND #4 BETWEEN 'A%' AND 'M%'
   ```

   If you use abbreviations for column names, ZDT/Db2 expands them to the full column name when you press Enter. If you omit the keyword WHERE, ZDT/Db2 inserts it when you press Enter.

   The Column Selection/Edit panel now looks like this:

Figure 33. Primary Column Selection/Edit panel showing row selection criteria

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)              Column Selection/Edit                Line 1 of 16
TABLE HFMUSER.EMP
------- Row Selection Criteria ---- (Use SQL/PF4 for full screen edit) -------
1 Sel: WORKDEPT = 'A00' AND LASTNAME BETWEEN 'A%' AND 'M%'              +

Cmd Seq SHEX CL# Column name          Data type(length)  Null Default  Order A/D
                ****  Top of data  ****
___ ___ S      1 EMPNO                CHARACTER(6)            None       ___   _
___ ___ S      2 FIRSTNME             VARCHAR(12)             None       ___   _
___ ___ S      3 MIDINIT              CHARACTER(1)            None       ___   _
___ ___ S      4 LASTNAME             VARCHAR(15)             None       ___   _
___ ___ S      5 WORKDEPT             CHARACTER(3)        Y   Null       ___   _
___ ___ S      6 PHONENO              CHARACTER(4)        Y   Null       ___   _
___ ___ S      7 HIREDATE             DATE                Y   Null       ___   _
___ ___ S      8 JOB                  CHARACTER(8)        Y   Null       ___   _
___ ___ S      9 EDLEVEL              SMALLINT            Y   Null       ___   _
___ ___ S     10 SEX                  CHARACTER(1)        Y   Null       ___   _
___ ___ S     11 BIRTHDATE            DATE                Y   Null       ___   _
___ ___ S     12 SALARY               DECIMAL(9,2)        Y   Null       ___   _
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=SQL       F5=RFind     F6=RunTemp
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

2. Press the Exit function key (F3).

ZDT/Db2 displays the selected rows:

Figure 34. Example showing selected columns

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                  Table Edit                    3 rows fetched
TABLE HFMUSER.EMP                                               Format TABL
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+---->
000000 ****  Top of data  ****
000001 000110 VINCENZO<    G       LUCCHESI<       A00      3490    16.05.1958
000002 200010 DIAN<        J       HEMMINGER<      A00      3978    01.01.1965
000003 000010 CHRISTINE<   I       HAAS<           A00      3978    01.01.1965
000004 ****  End of data  ****









Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Using the SQL Edit session on page 95
Scrolling the Row Selection Criteria field on page 94
Expanding the Row Selection Criteria field on page 95

Related references

Row Selection Criteria panel on page 799

Scrolling the Row Selection Criteria field

To scroll the **Row Selection Criteria** entry field on the Column Selection/Edit panel, you can use the:

- Right function key (F11) and Left function key (F10), or
- RIGHT and LEFT primary commands

In either case, you must position the cursor on the **Row Selection Criteria** entry field before pressing either function key, or pressing Enter to issue either primary command.

The scroll indicator (+ or -) to the right of the expression indicates which directions you can scroll:

+

Entry field scrollable to the right

-

Entry field scrollable to the left

To control how far you scroll, enter a scroll amount in the **Scroll** field:

Scroll amount

Scrolls…

MAX

To the first or last position of the expression

PAGE

One viewable width of data at a time

HALF

Half a viewable width of data at a time

DATA

One character less than a viewable width of data at a time

CSR

When scrolling to the right, ZDT/Db2 positions the field with the character at the cursor being moved to the leftmost position in the viewable expression.

When scrolling to the left, ZDT/Db2 positions the field with the character at the cursor being moved to the rightmost position in the viewable expression.

nnnn

nnnn characters at a time

To change the scroll amount, enter a new value into the **Scroll** field. If you enter `MAX`, ZDT/Db2 restores the scroll amount to its previous value after the operation. If you enter any other scroll amount, it remains in effect until you change it again.

Related tasks

Row selection by column on page 86
Row selection criteria in freeform style on page 92

Related references

RIGHT primary command on page 996
LEFT primary command on page 971

Expanding the Row Selection Criteria field

You can expand the **Row Selection Criteria** entry field by:

- Entering the command SQL, or pressing the SQL function key (F4), to display an ISPF edit session containing the SQL expression
- Using the EXPAND command:
    1. Type `EXPAND` on the command line
    2. Position the cursor on the **Row Selection Criteria** entry field
    3. Press Enter.

       ZDT/Db2 displays an expanded entry panel that allows you to enter row selection criteria up to 32,767 bytes long.

Related tasks

Row selection by column on page 86
Row selection criteria in freeform style on page 92
Using the SQL Edit session on page 95

Related references

Column Selection/Edit panel on page 497
EXPAND primary command on page 949

Using the SQL Edit session

To start an SQL Edit session, use the SQL primary command. The Edit session contains the SQL statement that ZDT/Db2 is using to retrieve data from Db2®.

You can use the SQL Edit session to change the row selection criteria (WHERE clause), before retrieving data from Db2®.

If you use the SQL primary command from within an editor session, ZDT/Db2 displays a panel such as that shown in Figure 35: SQL Edit session: SQL command issued from within an editor session on page 96.

Figure 35. SQL Edit session: SQL command issued from within an editor session

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
EDIT       SYS02203.T120510.RA000.HFMUSER.R0113662         Columns 00001 00072
****** ***************************** Top of Data ******************************
=NOTE= All changes made during this Edit session will be discarded at the end
=NOTE= of the Edit session.
=NOTE= Use the CREATE or REPLACE commands to take a copy of the SQL.
000001 SELECT "EMPNO", "FIRSTNME", "MIDINIT", "LASTNAME", "WORKDEPT",
000002        "PHONENO", "HIREDATE", "JOB", "EDLEVEL", "SEX", "BIRTHDATE",
000003        "SALARY", "BONUS", "COMM"
000004 FROM   "DSN8810"."EMP"
000005 WHERE  "WORKDEPT" = 'A00' AND "LASTNAME" BETWEEN 'A' AND 'M'
000006 WITH   UR
000007 FOR    FETCH ONLY
****** **************************** Bottom of Data ****************************




Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

If you are editing a template (before displaying data) and you use the SQL primary command (or press the SQL function key (F4)), ZDT/Db2 displays a panel such as that shown in Figure 36: SQL Edit session: SQL command issued when editing a template on page 96.

Figure 36. SQL Edit session: SQL command issued when editing a template

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       SYS02248.T145911.RA000.HFMUSER.R0103542         Columns 00001 00072
****** ***************************** Top of Data ******************************
=NOTE= Only changes to the WHERE clause will be recognized at the end of the
=NOTE= Edit session.  All other changes will be ignored.
=NOTE= Use the CREATE or REPLACE commands to take a copy of the SQL.
=NOTE=
=NOTE= Columns for "DSN8810"."EMP"
=NOTE=     Cl# Column name         Data type(length)   Null
=NOTE=     1   EMPNO               CHARACTER(6)
=NOTE=     2   FIRSTNME            VARCHAR(12)
=NOTE=     3   MIDINIT             CHARACTER(1)
=NOTE=     4   LASTNAME            VARCHAR(15)
=NOTE=     5   WORKDEPT            CHARACTER(3)        Y
=NOTE=     6   PHONENO             CHARACTER(4)        Y
=NOTE=     7   HIREDATE            DATE                Y
=NOTE=     8   JOB                 CHARACTER(8)        Y
=NOTE=     9   EDLEVEL             SMALLINT            Y
=NOTE=     10  SEX                 CHARACTER(1)        Y
=NOTE=     11  BIRTHDATE           DATE                Y
Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

You can use the ISPF editor commands CREATE and REPLACE to take a copy of the SQL statement into another, permanent, data set.

Column details for the Db2® object might be shown as note lines (**=NOTE=**) at the top of the ISPF edit session. These are only displayed if the WHERE clause can be changed.

Related references

SQL primary command on page 1005

Making changes

The note lines (**=NOTE=**) at the top of the SQL Edit session indicate whether you can make changes to the SQL statement. When data is displayed in a Z Data Tools/Db2 editor session, you cannot change the SQL statement without first ending the current editor session. To interactively test and change SQL statements, use Prototyping: Enter (option 4.3), or use the **Re-edit template** option on the Db2® View or Db2® Edit panel.

You can use ZDT/Db2 reference numbers as abbreviations for the column names.

When you exit the SQL Edit session by pressing the Exit function key (F3), ZDT/Db2 converts ZDT/Db2 reference numbers to the corresponding Db2® column name, and validates the statement. If the validation fails, ZDT/Db2 redisplays the SQL statement with an error message.

Note:

1. You cannot change any part of the SQL statement before the WHERE keyword. This includes the list of columns in the SELECT clause (see the following note item).
2. All columns in the table are included in the SELECT clause (following the SELECT keyword) of the SQL statement, even if you have edited the template to deselect some columns. If you deselect a column, it causes ZDT/Db2 not to display that column; it does not affect the SQL command used by ZDT/Db2 to retrieve the data from Db2®.
3. The maximum length for the WHERE clause that can be specified in the ISPF editor session displayed by the SQL command is 32767 characters.

Related tasks

Ending a Z Data Tools/Db2 editor session on page 133
Entering, executing, and explaining SQL statements on page 424
Selecting rows on page 86
Selecting columns on page 99

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616
SQL primary command on page 1005

Saving changes

You can only save any changes you make in the SQL Edit session if the WHERE clause can be changed, and the resultant SQL statement is acceptable to Db2®.

When you have made the required changes, press the Exit function key (F3) to redisplay the Column Selection/Edit panel. The row selection criteria section of the panel shows the changes you made to the WHERE clause in the Edit session.

To return to the Column Selection/Edit panel without saving your changes, press the Cancel function key (F12).

Related references

Column Selection/Edit panel on page 497

Long WHERE clauses

You can use the ZDT/Db2 SQL Edit session to enter long WHERE clauses (subject to any limits imposed by your host Db2® system). See the *DB2 for z/OS SQL Reference*.

If the SQL expression is too large to be displayed in the **Row Selection Criteria** section of the Column Selection/Edit panel, ZDT/Db2 displays as much of the expression as possible. In this situation, if you make any changes to the expression on the Column Selection/Edit panel, ZDT/Db2 displays an ISPF Edit session containing the complete SQL expression.

The maximum length for the WHERE clause that can be specified in the ISPF editor session displayed by the SQL command is 32767 characters.

Related references

Column Selection/Edit panel on page 497

Manipulating columns

ZDT/Db2 offers a number of ways to manipulate columns within rows. Using the Column Selection/Edit panel, you can:

- Select columns
- Change the order in which columns are displayed
- Change column headings
- Access column attributes panels to change individual column attributes

Related tasks

Selecting columns on page 99
Changing the sequence of displayed or printed data on page 101
Changing the column heading on page 105
Changing column attributes on page 104

Related references

Column Selection/Edit panel on page 497

Selecting columns

To select which columns are displayed in a Z Data Tools/Db2 editor session, use the Column Selection/Edit panel. Selected columns are displayed with an *"S"* in the **S** column. To toggle between selecting and deselecting a column, enter `s` in the **Cmd** column of the Column Selection/Edit panel.

By default, ZDT/Db2 selects all of the columns in the Db2® object for display, in the order that they are defined to Db2®. The initial display shows all columns as selected, with the exception of any ROWID column.

Note: Deselecting a column does not alter the SQL statement that ZDT/Db2 uses to retrieve data from Db2®. In a Z Data Tools/Db2 editor session, ZDT/Db2 always generates an SQL statement that specifies every column in the Db2® object.

Toggling the selection status of more than one column at a time

You can toggle between selecting or deselecting more than one column at a time by entering one of the following selection commands:

SS

Selects or deselects a block of columns.

Enter SS twice: first in the **Cmd** column at the start of the block you want to select or deselect, and again at the end of the block. ZDT/Db2 toggles the selection status of each column in the selected block.

S*n*

Selects or deselects *n* columns.

Enter S*n* in the **Cmd** field of the first column you want to select or deselect. ZDT/Db2 toggles the selection status of each column, starting from the first column selected for *n* columns.

S*

Selects or deselects a block of columns, starting from the first column selected and then all subsequent columns.

Enter S* in the **Cmd** field of the first column you want to select or deselect. ZDT/Db2 toggles the selection status of each column, starting from the first column selected to the last column.

Note: If every column in a template is deselected, ZDT/Db2 selects every column in the template for processing.

Related tasks

ROWID columns on page 107

Related references

Column Selection/Edit panel on page 497

Holding columns on the display when scrolling

You can hold columns so that they are always displayed regardless of how far you scroll.

You can hold columns so that, if you scroll left or right in TABL display format, those columns are always displayed on the left of the panel (regardless of how far you scroll).

To hold a column, enter `H` in the **Cmd** column of the Column Selection/Edit panel. To free the column, enter `H` again in the **Cmd** column.

If you hold a column, *"H"* is displayed in the H column on the Column Selection/Edit panel, and *"1"* is displayed in the **Seq** column to show it is the first held column. If you hold a second column, *"2"* is displayed in the **Seq** column to show it is the second held column, and so on.

Note:

1. The sequence numbers shown in the **Seq** column for held columns are independent of any sequence number you might enter to change the sequence of displayed columns. Held columns are always displayed as the leftmost columns on the panel (in the order that you held them when editing the template) and take priority (as far as the left-to-right sequence in which columns are displayed is concerned) over other columns.
2. If you hold a number of columns, there might be little or no room left on the display for non-held columns. This is a particular problem if using an 80-byte display. In this situation you might not be able to scroll right.

Toggling the hold status of multiple columns

You can hold or free more than one column at a time by entering one of the following commands in the **Cmd** column of the Column Selection/Edit panel:

HH

Holds or frees a block of columns.

Enter HH twice: first in the **Cmd** column at the start of the block of columns you want to hold or free, and again at the end of the block. ZDT/Db2 toggles the hold status of each column in the selected block.

H*n*

Holds or frees *n* columns.

Enter H*n* in the **Cmd** field of the first column you want to hold or free. ZDT/Db2 toggles the hold status of each column, starting from the first column selected for *n* columns.

Related tasks

Changing the sequence of displayed or printed data on page 101

Related references

Column Selection/Edit panel on page 497

How to use this reference chapter

This chapter lists the ZDT/Db2 batch functions in alphabetical order.

The following subsections are supplied, where relevant, for each function:

Purpose

A short summary of what you can do with the function.

Usage notes

A more detailed explanation of how to use the function.

Syntax

A syntax diagram, followed by a parameter list. The parameter list describes the parameter, and shows (where appropriate) its default and other possible values. If you specify parameters that are not relevant to the function, Z Data Tools ignores them. If you specify the same parameter more than once, Z Data Tools uses the first value that you specify.

Batch example

A listing of a sample batch job.

Return codes

Lists the return codes that are specific to the function.

Related functions

Other functions that are related in some way (for example, the export and import functions can be used to transfer data between Db2® systems)

Changing the sequence of displayed or printed data

You can edit a template to change:

- the order in which *columns* are displayed or printed
- the sequence in which *rows* are displayed or printed

Related tasks

Changing the sequence of displayed or printed columns on page 101
Changing the sequence of displayed or printed rows on page 103

Changing the sequence of displayed or printed columns

By default, ZDT/Db2 displays or prints columns in the sequence they are defined in the Db2® catalog. This is the same sequence that the column names are displayed in the Column Selection/Edit panel.

To change the order in which columns are displayed or printed, use the **Seq** field on the Column Selection/Edit panel to specify an alternative order.

Note: The **Seq** field on the Column Selection/Edit panel affects only the sequence in which ZDT/Db2 displays or prints columns. It has no effect on the sequence in which columns are exported when you use the Export utility function (3.7).

For example, on the Column Selection/Edit panel shown in Figure 37: Primary Column Selection/Edit panel showing column sequence selection on page 102, to make the JOB column be shown first, then the HIREDATE column (followed by the remaining columns):

1. In the **Seq** column against the JOB column, type `1`.
2. In the **Seq** column against the HIREDATE column, type `2`.

   The Column Selection/Edit panel now looks like this:

   Figure 37. Primary Column Selection/Edit panel showing column sequence selection

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)              Column Selection/Edit                Line 1 of 16
TABLE HFMUSER.EMP
------- Row Selection Criteria ---- (Use SQL/PF4 for full screen edit) -------
1 Sel: ____________________________________________________________________  +

Cmd Seq SHEX CL# Column name          Data type(length)  Null Default  Order A/D
                ****  Top of data  ****
___ ___ S      1 EMPNO                CHARACTER(6)            None       ___   _
___ ___ S      2 FIRSTNME             VARCHAR(12)             None       ___   _
___ ___ S      3 MIDINIT              CHARACTER(1)            None       ___   _
___ ___ S      4 LASTNAME             VARCHAR(15)             None       ___   _
___ ___ S      5 WORKDEPT             CHARACTER(3)        Y   Null       ___   _
___ ___ S      6 PHONENO              CHARACTER(4)        Y   Null       ___   _
___ __2 S      7 HIREDATE             DATE                Y   Null       ___   _
___ __1 S      8 JOB                  CHARACTER(8)        Y   Null       ___   _
___ ___ S      9 EDLEVEL              SMALLINT            Y   Null       ___   _
___ ___ S     10 SEX                  CHARACTER(1)        Y   Null       ___   _
___ ___ S     11 BIRTHDATE            DATE                Y   Null       ___   _
___ ___ S     12 SALARY               DECIMAL(9,2)        Y   Null       ___   _
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=SQL       F5=RFind     F6=RunTemp
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

3. Press the RunTemp function key (F6).

The data is displayed showing the revised order of the columns:

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Table Browse                    42 rows fetched
TABLE HFMUSER.EMP                                                 Format TABL
JOB      HIREDATE   EMPNO  FIRSTNME     MIDDLE INITIAL LASTNAME        WORKDEP
#8       #7         #1     #2           #3             #4              #5
CH(8)    DATE       CH(6)  VARCHAR(12)  CHARACTER(1)   VARCHAR(15)     CH(3)
<---+--> <---+----> PU--+> <---+----1-> -              <---+----1----> <-NF
****  Top of data  ****
PRES     01/01/1965 000010 CHRISTINE<   I              HAAS<           A00
MANAGER  10/10/1973 000020 MICHAEL<     L              THOMPSON<       B01
MANAGER  04/05/1975 000030 SALLY<       A              KWAN<           C01
MANAGER  08/17/1949 000050 JOHN<        B              GEYER<          E01
MANAGER  09/14/1973 000060 IRVING<      F              STERN<          D11
MANAGER  09/30/1980 000070 EVA<         D              PULASKI<        D21
MANAGER  08/15/1970 000090 EILEEN<      W              HENDERSON<      E11
MANAGER  06/19/1980 000100 THEODORE<    Q              SPENSER<        E21
SALESREP 05/16/1958 000110 VINCENZO<    G              LUCCHESI<       A00
CLERK    12/05/1963 000120 SEAN<                       O'CONNELL<      A00
ANALYST  07/28/1971 000130 DOLORES<     M              QUINTANA<       C01
ANALYST  12/15/1976 000140 HEATHER<     A              NICHOLLS<       C01
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Copying data to a VSAM or QSAM file on page 279

Related references

Column Selection/Edit panel on page 497

Changing the sequence of displayed or printed rows

To change the order in which rows are displayed or printed, use the **Order** field and, optionally, the **A/D** field on the Column Selection/Edit panel. For the most significant column, type “1” in the **Order** field; for the next most significant column, type “2” in the **Order** field, and so on. If you use the **Order** field, rows are displayed in ascending sequence by default. To display the rows in descending sequence, type “D” in the **A/D** field.

If you use either the **Order** or **A/D** fields on the Column Selection/Edit panel, ZDT/Db2 adds an ORDER BY clause to the SQL SELECT statement used to retrieve the data from Db2®.

For example, on the Column Selection/Edit panel shown in Figure 38: Primary Column Selection/Edit panel showing row sequence selection on page 104, the **Order** and **A/D** fields specify rows to be shown in the sequence LASTNAME (ascending) within WORKDEPT (descending).

Figure 38. Primary Column Selection/Edit panel showing row sequence selection

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)              Column Selection/Edit                Line 1 of 16
TABLE HFMUSER.EMP
------- Row Selection Criteria ---- (Use SQL/PF4 for full screen edit) -------
1 Sel: ____________________________________________________________________  +

Cmd Seq SHEX CL# Column name          Data type(length)  Null Default  Order A/D
                ****  Top of data  ****
___ ___ S      1 EMPNO                CHARACTER(6)            None       ___   _
___ ___ S      2 FIRSTNME             VARCHAR(12)             None       ___   _
___ ___ S      3 MIDINIT              CHARACTER(1)            None       ___   _
___ ___ S      4 LASTNAME             VARCHAR(15)             None       __2   A
___ ___ S      5 WORKDEPT             CHARACTER(3)        Y   Null       __1   D
___ ___ S      6 PHONENO              CHARACTER(4)        Y   Null       ___   _
___ ___ S      7 HIREDATE             DATE                Y   Null       ___   _
___ ___ S      8 JOB                  CHARACTER(8)        Y   Null       ___   _
___ ___ S      9 EDLEVEL              SMALLINT            Y   Null       ___   _
___ ___ S     10 SEX                  CHARACTER(1)        Y   Null       ___   _
___ ___ S     11 BIRTHDATE            DATE                Y   Null       ___   _
___ ___ S     12 SALARY               DECIMAL(9,2)        Y   Null       ___   _
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=SQL       F5=RFind     F6=RunTemp
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

When the data is displayed, it appears in the revised sequence:

Figure 39. Table Browse panel

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                  Table Browse                    42 rows fetched
TABLE HFMUSER.EMP                                              Format TABL
EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE   JOB
#1     #2           #3      #4              #5       #6      #7         #8   +
CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE       CH(8)
PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+----> <---+-
****  Top of data  ****
200340 ROY<         R       ALONZO<         E21      5698    05/05/1947 FIELDR
000340 JASON<       R       GOUNOT<         E21      5698    05/05/1947 FIELDR
000330 WING<                LEE<            E21      2103    02/23/1976 FIELDR
000320 RAMLAL<      V       MEHTA<          E21      9990    07/07/1965 FIELDR
000100 THEODORE<    Q       SPENSER<        E21      0972    06/19/1980 MANAGE
200330 HELENA<              WONG<           E21      2103    02/23/1976 FIELDR
000090 EILEEN<      W       HENDERSON<      E11      5498    08/15/1970 MANAGE
000290 JOHN<        R       PARKER<         E11      4502    05/30/1980 OPERAT
000280 ETHEL<       R       SCHNEIDER<      E11      8997    03/24/1967 OPERAT
200280 EILEEN<      R       SCHWARTZ<       E11      8997    03/24/1967 OPERAT
000310 MAUDE<       F       SETRIGHT<       E11      3332    09/12/1964 OPERAT
000300 PHILIP<      X       SMITH<          E11      2095    06/19/1972 OPERAT
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related references

Column Selection/Edit panel on page 497

Changing column attributes

Attributes are used to determine how to display, print, or create a column.

To change a column attribute, use the Column Selection/Edit panel and enter `E` in the **Cmd** field against one or more columns. For each column you select, ZDT/Db2 displays a Column Attributes panel.

The attributes you can change are:

- Column heading
- Output width
- Whether to show leading zeros (numeric columns)
- Values to use for creating new columns

Note:

1. For numeric, DATE, TIME, and TIMESTAMP columns, you can change the column heading as for an alphanumeric column.
2. For DATE, TIME, and TIMESTAMP columns, you can change the output width as for an alphanumeric column.
3. For very long columns, change the display width to a smaller value to see more data on the display.

Related tasks

Changing the attributes for an alphanumeric column on page 105
Changing the attributes for a numeric column on page 107

Related references

Column Selection/Edit panel on page 497
Column Attributes panel (alphanumeric) on page 484
Column Attributes panel (numeric) on page 490
Column Attributes panel (DATE) on page 489
Column Attributes panel (TIME) on page 494
Column Attributes panel (TIMESTAMP) on page 495

Changing the attributes for an alphanumeric column

Enter `E` in the **Cmd** field for an alphanumeric column to display the Column Attributes panel. Use this panel to change the heading or width of an alphanumeric column, or to set the pattern if you are creating alphanumeric columns.

Changing the column heading

By default, ZDT/Db2 uses the column name defined in the Db2® catalog as the column heading. You can change this by editing the template and specifying an alternative column heading in the **Heading** entry field.

The column heading might change the width of the display column. For example, in Figure 39: Table Browse panel on page 104, see how the display changes if you change the heading for the MIDINIT column to the longer heading, *"MIDDLE INITIAL"*:

1. Enter the TEDIT primary command.

   ZDT/Db2 displays the Column Selection Edit panel.

2. On the line for the MIDINIT column, enter `E` in the **Cmd** field.

   ZDT/Db2 displays the Column Attributes panel for the MIDINIT column.

3. Type the new heading, `MIDDLE INITIAL`, in the **Heading** entry field.
4. Press the Exit function key (F3).

   ZDT/Db2 displays the Column Selection/Edit panel.

5. Press the Exit function key (F3) again.

   ZDT/Db2 displays the Template Save panel.

6. Press the RunTemp function key (F6).

   ZDT/Db2 displays the data with the new heading.

Figure 40: Table Browse panel showing width of displayed column changed due to longer heading on page 106 shows how the change of heading has changed the width of the third column from what it was when originally displayed in Figure 39: Table Browse panel on page 104.

Figure 40. Table Browse panel showing width of displayed column changed due to longer heading

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
 ZDT/Db2 (DFG2)                  Table Browse                    42 rows fetched
 TABLE HFMUSER.EMP                                              Format TABL
 EMPNO  FIRSTNME     MIDDLE INITIAL LASTNAME        WORKDEPT PHONENO HIREDATE
 #1     #2           #3             #4              #5       #6      #7
 CH(6)  VARCHAR(12)  CHARACTER(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
 PU--+> <---+----1-> -              <---+----1----> <-NF     <-->    <---+---->
 ****  Top of data  ****
 200340 ROY<         R              ALONZO<         E21      5698    05/05/1947
 000340 JASON<       R              GOUNOT<         E21      5698    05/05/1947
 000330 WING<                       LEE<            E21      2103    02/23/1976
 000320 RAMLAL<      V              MEHTA<          E21      9990    07/07/1965
 000100 THEODORE<    Q              SPENSER<        E21      0972    06/19/1980
 200330 HELENA<                     WONG<           E21      2103    02/23/1976
 000090 EILEEN<      W              HENDERSON<      E11      5498    08/15/1970
⋮
 Command ===> _____________________________________________________ Scroll PAGE
  F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Changing the output width

To change the output width, enter the new width into the **Output width** entry field.

The output width is the number of character positions used by view, edit, browse, and print panels to show the column in TABL display or print format. The default output width is the maximum of the number of characters needed to show the column heading (or column name, if no heading is specified), and the number of characters needed to show the value of the column, as determined from the column definition in the copybook.

The minimum width is six characters. The maximum is 30 or the column width + 10 (whichever is the greater).

Related references

Column Attributes panel (alphanumeric) on page 484

Changing the attributes for a numeric column

Enter `E` in the **Cmd** field for a numeric column to display the numeric Column Attributes panel. Use this panel to change the heading or display attributes of a numeric column, or to set the pattern if you are creating numeric columns.

For numeric columns, you change the column heading as for an alphanumeric column. See Changing the attributes for an alphanumeric column on page 105.

Changing the output width

To change the output width, enter the new width into the **Output width** column. The output width is the number of character positions used by view, edit, browse, and print panels to show the column in TABL display or print format. The default output width is the maximum of the number of characters needed to show the column heading (or column name, if no heading is specified), and the number of characters needed to show the value of the column, as determined from the column definition in the copybook, including a sign character and decimal point.

The allowable widths are 6 to 32 characters.

Showing leading zeros

To show leading zeros, enter `YES` into the **Leading zeros** column. The default setting is NO (suppress leading zeros). Figure 41: An example display with leading zeros shown for a column (SALARY) on page 107 shows the leading zeros shown for SALARY (but not for the other two numeric columns shown).

Figure 41. An example display with leading zeros shown for a column (SALARY)

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
 ZDT/Db2 (DFG2)                    Table Browse          42 rows fetched
 TABLE HFMUSER.EMP                                            Format TABL
 JOB       EDLEVEL SEX    BIRTHDATE       SALARY        BONUS        COMM
 #8             #9 #10    #11               #12          #13          #14
 CH(8)       SMINT CH(1)  DATE          DEC(9,2)     DEC(9,2)     DEC(9,2)
 <---+--->  <---+> -      <---+----> <---+----1> <---+----1> <---+----1>
 ****  Top of data  ****
 PRES           18 F      08/14/1933  0052750.00     1000.00     4220.00
 MANAGER        18 M      02/02/1948  0041250.00      800.00     3300.00
 MANAGER        20 F      05/11/1941  0038250.00      800.00     3060.00
 MANAGER        16 M      09/15/1925  0040175.00      800.00     3214.00
 MANAGER        16 M      07/07/1945  0032250.00      600.00     2580.00
 MANAGER        16 F      05/26/1953  0036170.00      700.00     2893.00
 MANAGER        16 F      05/15/1941  0029750.00      600.00     2380.00
⋮
 Command ===> _____________________________________________________ Scroll PAGE
  F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related references

Column Attributes panel (numeric) on page 490

ROWID columns

You can use ROWID columns to create a unique index for tables that have no obvious unique index column or columns.

After a row has been inserted, you cannot update the data in a ROWID column. You can only specify a ROWID value on insert operations for ROWID columns defined as GENERATED BY DEFAULT. See the *DB2 for z/OS SQL Reference*.

When ZDT/Db2 generates the template for a table containing a ROWID column, the ROWID column is deselected and is not initially displayed in a Z Data Tools/Db2 editor session.

Specifying scrambling options

Scrambling data allows you to create test data based on production (or "live") data, but with the ability to change the values of certain columns.

When you use the Copy Utility (option 3.3), Import Utility (option 3.6), or Export Utility (option 3.7), you can choose to scramble some or all of the columns being copied. You specify the columns you want scrambled by marking those columns for scrambling in the output template.

You mark a column for scrambling by setting the scrambling options for that column in the template.

You can specify or change scrambling options on the Column Attributes panel. The panel comes in two flavors: one for alphanumeric columns and the other for numeric columns.

Selecting a column for specifying scrambling options on page 108
Scrambling data on page 292
Column Attributes panel (alphanumeric) on page 484
Column Attributes panel (numeric) on page 490
Value List Edit panel on page 917
Scramble Exit Specification panel on page 814

Selecting a column for specifying scrambling options

You select a column from the Column Selection/Edit panel. To select a column for specifying scrambling options, enter `E` in the **Cmd** field adjacent to the required column. You can enter E against as many columns as you like. For each column you select, Z Data Tools displays a Column Attributes panel.

After you have updated the scrambling options for a column (or any of the other column attributes), when you return to the Column Selection/Edit panel you will see an "E" for that column in the E column (under the "SHEX" heading) indicating that the attributes for that column have changed.

The scrambling options you can specify are:

- Scramble type
- Value option
- Value input column
- Value output column
- Minimum and maximum values for a numeric range
- Value data set name

The scrambling options determine how the input data is scrambled (if at all) when it is copied, imported or exported.

Related topics

Specifying the scramble type on page 109
Specifying the value option on page 110
Specifying value input and out columns on page 110
Specifying range values on page 111
Specifying a value data set name on page 111
Specifying and editing a value list on page 112

Specifying the scramble type

The **Scramble Type** option determines how data is scrambled when it is copied, imported, or exported.

Set the scramble type to one of these values:

Blank

Data is not scrambled.

1 ("Random")

Data scrambled to produce a random output value on each invocation.

2 ("Repeatable")

Data scrambled to produce the same output value on each invocation.

3 ("Translate")

Data scrambled using input and output values held in the value data set (specified in the **Dsn** field).

This option requires you to select the **Value** option with an input and output column (**In** and **Out**) and value data set name (**Dsn**). All records in the translate data set are loaded into memory for the copy operation. The input and output columns provide the location of the input and output field values as stored in the data set. Their lengths are determined by the respective input and output columns that have been mapped for the copy, import, or export operation. If a matching input column value is found, then the corresponding output column value is obtained from the matching record. If no match is found, then the column is initialized to zero, or the value determined by the data create attributes.

4 ("Exit")

Data scrambled using a scrambling exit (as specified on a separate panel). For details about writing a scrambling exit, see the *Z Data Tools Customization Guide*.

Related topics

Specifying the value option on page 110
Specifying value input and out columns on page 110

Specifying range values on page 111
Specifying a value data set name on page 111

Specifying the value option

The **Value** option and corresponding fields control the output of the scrambling process.

The way in which you select, or specify, this option differs according to whether the associated column is alphanumeric or numeric:

- For alphanumeric column, do one of these:
  - Deselect the **Value** option by entering a blank.
  - Select the **Value** option by entering a "/".
- For numeric columns, do one of these:
  - Deselect the **Value** option by entering a blank.
  - Select a range by entering "1".
  - Select the **Value** option by entering "2".

Value option selected ("/")

Selecting this option allows you to provide:

- **A translate data set.** This applies when you select **Scramble Type** 3 ("Translate") with input and output columns and a value data set.
- **A lookup data set.** This applies when you select a **Scramble Type** of 1 ("Random") or 2 ("Repeatable") along with a value data set name. The data set is loaded into memory and the output value is randomly or repeatably selected from one of the loaded records. The output column determines the location of the value that is selected for this column.
- **A value list.** This applies when you select a **Scramble Type** of 1 ("Random") or 2 ("Repeatable") *without* a value data set name and allows you to enter the selection values for this column on the panel displayed when you press Enter. One of the values you provide is randomly or repeatably selected during a copy, import, or export operation. The values you enter are stored in the template. Value lists are described in more detail later.

Value option deselected (blank)

Deselecting this option scrambles the input column to produce a random or repeatable output value. **Scramble Type** must be set to 1 ("Random") or 2 ("Repeatable").

Specifying value input and out columns

The input column field (**In**) defines the start location of the input field value on the value data set and is used when the translate process is run during a copy, import, or export operation to match the input column with a value on the value data set. The length of the field is set to the length of the input column that is mapped to this column during the copy, import, or export process.

Note: This value is only required when you select **Scramble Type** 3 ("Translate").

The output column field **Out** defines the start location of the output field value on the value data set and is used in these ways during a copy operation:

- **Translate.** If an input column value is matched on the value data set, then the corresponding output value is used.
- **Random** or **Repeatable**. The input value is used to randomly, or repeatably, select an output value from the value data set.

Note:

This value is required when you select **Scramble Type** 3 ("Translate").

If you select **Scramble Type** 1 ("Random") or 2 ("Repeatable"), and you have also selected the **Value** option, then the output column defaults to 1 if you have specified a value data set name (**Dsn**).

Specifying range values

When you are specifying scramble options for a numeric column, you can specify a range of values from which the resultant scrambled value is to be selected.

You specify the minimum value of the range in the **Min** field and the maximum value of the range in the **Max** field.

You can only select the range option with random and repeatable scrambling. The output value is randomly or repeatably selected from a number within the specified range.

Specifying a value data set name

The output data set name field **Dsn** field defines the value data set. It can be any cataloged sequential, partitioned or VSAM data set containing data that is used to determine the output column value during a copy, import, or export process.

If you select **Scramble Type** 3 ("Translate"), then the data set must contain the input and output values in the locations provided in the input and output columns.

If you select **Scramble Type** 1 ("Random") or 2 ("Repeatable"), then the data set must contain the output value in the locations provided in the output column.

Notes:

1. This field is required when you select **Scramble Type** 3 ("Translate"). If you select **Scramble Type** 1 ("Random") or 2 ("Repeatable"), and also select the **Value** option, then you can optionally provide a data set name. If you leave this field blank having selected the **Value** option, you are prompted to enter the value list to be stored in the template.
2. The value data set record length must be greater than or equal to the field length.

Specifying and editing a value list

A value list is a list of values which is used in conjunction with the scrambling options to select a value with which a column is populated during a copy, import, or export operation. When the output column is being populated, one of the values in this list is selected. Scrambling options combine the originating value with the random or repeatable seed to select a value from the list.

For random and repeatable scrambling, you can specify a value list inline by leaving the value data set field (**Dsn**) blank and selecting the **Value** option.

To edit the related value list, select the **Value** option and press Enter. ZDT/Db2 displays the Value List Edit panel.

The values you provide must be valid for the corresponding data type. Deleting a value list turns off the value list option on the previous panel.

To enter hexadecimal values, use the X'hhhh' format:

```
x'C1C2C3'
```

To provide leading blanks or a blank value, use a quoted string:

```
'   SMITH'
'        '
```

Blank value lines (no quotation marks) are ignored.

Use the FIND and LOCATE commands to bring a value containing or starting with a given string to the current line.

You can use prefix commands to copy, insert, move, repeat or delete lines.

To save the values, press F3. To ignore changes, press Cancel.

Related topics

Value List Edit panel on page 917

Mapping data

Mapping tells ZDT/Db2 how you want columns transferred from an old table to a new table. You can set up mapping so that, in the extreme case, a column is copied to a column of a different name, with a different length, and with a different, but compatible, data type. Most mappings are more straightforward.

Other types of mapping supported by ZDT/Db2 include:

- Mapping the columns of one table to another.

  You can use this type of mapping with the ZDT/Db2 Copy utility.
- Mapping the columns of a table to the fields in a file.

  You can use this type of mapping with the ZDT/Db2 Export utility and the Db2® UNLOAD utility.
- Mapping the fields in a file to the columns of a Db2® table.

  You can use this type of mapping with the ZDT/Db2 Import utility and the Db2® LOAD utility.

Where you can map data

You can use templates to map data with the following ZDT/Db2 functions:

- Copy (option 3.3)
- Import (option 3.6)
- Export (option 3.7)
- Utilities (option 3.9)

Related tasks

Copying data from one Db2 object to another on page 246
Copying data from a VSAM or QSAM file on page 251
Copying data to a VSAM or QSAM file on page 279
LOAD utility on page 357

Letting ZDT/Db2 do the mapping

If you want, you can use ZDT/Db2 to set up the mappings. This happens automatically if you do not select the **Edit template mapping** on the Copy *"To"* panel. If no mapping exists in the *"To"* template, ZDT/Db2 performs a corresponding move by going through the following two steps:

1. Generates mappings based on column names. That is, columns in the *"From"* template are mapped to like-named columns in the *"To"* template. ZDT/Db2 provides mappings for as many columns as it can in this way.
2. For the remaining *"unmatched"* columns in the *"From"* template (if any), ZDT/Db2 generates mapping based on data attributes:
   - ZDT/Db2 compares the attributes of the first unmatched column in the *"From"* template with the attributes of the first unmatched column in the *"To"* template. If the attributes of the two columns are the same, ZDT/Db2 generates mapping between the two columns.
   - ZDT/Db2 repeats the preceding step with the next pair of unmatched columns in the two templates until either it has generated mapping for all the columns in the *"From"* template, or if it finds the attributes of the two columns do not match.

Related references

Copy Utility ("To") panel on page 521
Import Utility ("To") panel on page 720
Export Utility ("To") panel on page 701
LOAD from panel on page 731
UNLOAD Utility (Tables) "To" panel on page 911

Specifying your own mapping

If you want to set up your own mappings, indicate that you want to edit the *"To"* template by selecting the **Edit template mapping** option on the *"To"* panel for the relevant ZDT/Db2 function. If you select this option, ZDT/Db2 displays the Template Mapping panel (see Figure 42: Template Mapping panel on page 114).

Figure 42. Template Mapping panel

```
Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                Template Mapping

To          HFMUSER.EEMP
From        HFMUSER.EMP

Cmd CL# To Column name  Datatype  NI   CL# From Column name        Datatype  NI
       ****  Top of data  ****
___    1 EMPNO           CH(6)     NN |   1 EMPNO                  CH(6)     NN
___    2 FIRSTNME        VC(12)    NN |   2 FIRSTNME               VC(12)    NN
___    3 MIDINIT         CH(1)     NN |   3 MIDINIT                CH(1)     NN
___    4 LASTNAME        VC(15)    NN |   4 LASTNAME               VC(15)    NN
___    5 WORKDEPT        CH(3)        |   5 WORKDEPT               CH(3)
___    6 PHONENO         CH(4)        |   6 PHONENO                CH(4)
___    7 HIREDATE        DATE         |   7 HIREDATE               DATE
___    8 JOB             CH(8)        |   8 JOB                    CH(8)
___    9 EDLEVEL         SMINT        |   9 EDLEVEL                SMINT
___   10 SEX             CH(1)        |  10 SEX                    CH(1)
___   11 BIRTHDATE       DATE         |  11 BIRTHDATE              DATE
___   12 SALARY          DEC(9,2)     |  12 SALARY                 DEC(9,2)
___   13 BONUS           DEC(9,2)     |  13 BONUS                  DEC(9,2)
___   14 COMM            DEC(9,2)     |  14 COMM                   DEC(9,2)
S     15 RID             CH(4)        |
___   16 TSTAMP          TIMESTAMP    |
       ****  End of data  ****
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Note that, in Figure 42: Template Mapping panel on page 114, there is no *"From"* columns against the *"To"* columns RID and TSTAMP. This shows that, for these two columns in the *"To"* template, there is no corresponding column in the *"From"* template. When the table is copied, ZDT/Db2 creates new values in RID and TSTAMP.

Related references

Template Mapping panel on page 900
Copy Utility ("To") panel on page 521
Import Utility ("To") panel on page 720
Export Utility ("To") panel on page 701
LOAD from panel on page 731
UNLOAD Utility (Tables) "To" panel on page 911

Setting up mapping for a column

To set up or change the mapping for a column, enter `s` in the **Cmd** field against the column name. The From Column Mapping panel is displayed.

Figure 43. From Column Mapping panel

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
 ZDT/Db2 (DFG2)               From Column Mapping
 To Db2 object . . : DSN8810.EEMP
 From Db2 object . : DSN8810.EMP
 To column . . . . : #15 RID
 From column . . . :

 Sel   Cl#  From Column             Data Type(length)
            ****  Top of data  ****
 ___     D  Delete "From column"
 ___     1  EMPNO                  CHARACTER(6)
 ___     2  FIRSTNME               VARCHAR(12)
 ___     3  MIDINIT                CHARACTER(1)
 ___     4  LASTNAME               VARCHAR(15)
 ___     5  WORKDEPT               CHARACTER(3)
 S__     6  PHONENO                 CHARACTER(4)
 ___     7  HIREDATE                DATE
 ___     8  JOB                     CHARACTER(8)
 ___     9  EDLEVEL                SMALLINT
 ___    10  SEX                     CHARACTER(1)
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
  F9=Swap     F12=Cancel
```

In Figure 43: From Column Mapping panel on page 115, the information at the top of the panel shows you the current mapping. In the figure, the **To column** is RID. There is currently no **From column** shown because there is no column in the *"From"* template with the same name.

To *change* the *"From"* column, enter `s` against the new *"From"* column. The information at the top of the panel changes immediately to reflect the new mapping. For example, if you enter `s` against column #6 (PHONENO), the information at the top of the panel changes:

```
;
 ZDT/Db2 (DFG2)               From Column Mapping
 To Db2 object . . : DSN8810.EEMP
 From Db2 object . : DSN8810.EMP
 To column . . . . : #15 RID
 From column . . . : #6 PHONENO
```

and the Template Mapping panel also changes to show that PHONENO is mapped to RID:

```
;
 ZDT/Db2 (DFG2)                Template Mapping

 To          HFMUSER.EEMP
 From        HFMUSER.EMP

 Cmd CL# To Column name  Datatype  NI   CL# From Column name        Datatype  NI
         ****  Top of data  ****
 ___    1 EMPNO           CH(6)     NN |   1 EMPNO                  CH(6)     NN
:
 ___   12 SALARY          DEC(9,2)     |  12 SALARY                 DEC(9,2)
 ___   13 BONUS           DEC(9,2)     |  13 BONUS                  DEC(9,2)
 ___   14 COMM            DEC(9,2)     |  14 COMM                   DEC(9,2)
 ___   15 RID             CH(4)        |   6 PHONENO                CH(4)
 ___   16 TSTAMP          TIMESTAMP    |
         ****  End of data  ****
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
  F7=Up        F8=Down      F9=Swap     F12=Cancel
```

There is now an entry in the *"From"* column against RID.

To *remove* the current mapping for a *"To"* column, enter s against *""***Delete From column** on the From Column Mapping panel for that column. For example, the Template Mapping panel shown in Figure 42: Template Mapping panel on page 114 shows that the *"To"* column BONUS is mapped to the *"From"* column BONUS. To remove the mapping for BONUS (so that it is not mapped to any *"From"* column):

1. On the Template Mapping panel shown in Figure 42: Template Mapping panel on page 114, enter S in the **Cmd** field against column #13 (BONUS).

   The From Column Mapping panel is displayed as shown in Figure 44: From Column Mapping panel showing Delete "From column" selected on page 116.

2. Enter S against **Delete "From column"**.
   Figure 44. From Column Mapping panel showing **Delete "From column"** selected

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)               From Column Mapping
To Db2 object . . : DSN8810.EEMP
From Db2 object . : DSN8810.EMP
To column . . . . : #13 BONUS
From column . . . : #13 BONUS

Sel   Cl#  From Column             Data Type(length)
           ****  Top of data  ****
S__     D  Delete "From column"
___     1  EMPNO                  CHARACTER(6)
___     2  FIRSTNME               VARCHAR(12)
___     3  MIDINIT                CHARACTER(1)
___     4  LASTNAME               VARCHAR(15)
___     5  WORKDEPT               CHARACTER(3)
___     6  PHONENO                 CHARACTER(4)
        7  HIREDATE                 DATE
___     8  JOB                      CHARACTER(8)
___     9  EDLEVEL                 SMALLINT
___    10  SEX                      CHARACTER(1)
       11  BIRTHDATE                DATE
___    12  SALARY                   DECIMAL(9,2)
___    13  BONUS                    DECIMAL(9,2)
___    14  COMM                     DECIMAL(9,2)
           ****  End of data  ****
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F12=Cancel
```

3. Press Enter.

   The *"From column"* mapping information (fourth line on the From Column Mapping panel) for BONUS is displayed.

Figure 45. From Column Mapping panel showing **From column** mapping information deleted

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)               From Column Mapping
To Db2 object . . : DSN8810.EEMP
From Db2 object . : DSN8810.EMP
To column . . . . : #13 BONUS
From column . . . :

Sel   Cl#  From Column             Data Type(length)
           ****  Top of data  ****
___      D  Delete "From column"
___      1  EMPNO                  CHARACTER(6)
___      2  FIRSTNME               VARCHAR(12)
___      3  MIDINIT                 CHARACTER(1)
___      4  LASTNAME                VARCHAR(15)
___      5  WORKDEPT                CHARACTER(3)
___      6  PHONENO                 CHARACTER(4)
         7  HIREDATE                DATE
___      8  JOB                     CHARACTER(8)
___      9  EDLEVEL                 SMALLINT
___     10  SEX                     CHARACTER(1)
        11  BIRTHDATE               DATE
___     12  SALARY                  DECIMAL(9,2)
___     13  BONUS                   DECIMAL(9,2)
___     14  COMM                    DECIMAL(9,2)
           ****  End of data  ****
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F12=Cancel
```

4. Press the Exit function key (F3).

The Template Mapping panel is displayed showing the *"To"* column BONUS is now not mapped to any *"From"* column.

Figure 46. Template Mapping panel showing the "To" column BONUS not mapped to any "From" column

```
ZDT/Db2 (DFG2)               Template Mapping

To          HFMUSER.EEMP
From        HFMUSER.EMP

Cmd CL# To Column name  Datatype  NI   CL# From Column name        Datatype  NI
        ****  Top of data  ****
___    1 EMPNO           CH(6)     NN |   1 EMPNO                  CH(6)     NN
⋮
___   12 SALARY          DEC(9,2)     |  12 SALARY                 DEC(9,2)
___   13 BONUS           DEC(9,2)     |
___   14 COMM            DEC(9,2)     |  14 COMM                   DEC(9,2)
___   15 RID             CH(4)        |   6 PHONENO                CH(4)
___   16 TSTAMP          TIMESTAMP    |
        ****  End of data  ****
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Note: Db2® determines the default value for an unmapped column according to the definition of the column in the Db2® catalog.

Changing the attributes for a "To" column

To change the attributes for a *"To"* column, enter `E` against the column name on the Template Mapping panel (Figure 42: Template Mapping panel on page 114) to display the Column Attributes panel.

Related tasks

Changing column attributes on page 104

Creating data for a column

If no mapping exists for a *"To"* column, and ZDT/Db2 is unable to match it to a *"From"*, ZDT/Db2 creates data for the column.

Related tasks

Mapping data on page 112

Mapping using different templates

The mapping relationship is stored in the *"To"* template. The mapping information is the column number and the column name. If you reuse a template to copy, import, export, or load data, the mapping information is checked. ZDT/Db2 ensures that the column in the specified position has the specified name. If this is not the case, ZDT/Db2 indicates that the mapping has broken down, and gives you the option of either terminating the ZDT/Db2 function, or updating the mapping information in the *"To"* template.

This means that you can use a template that is different from the *"From"* template you used to set up the mapping, provided that the structure of the new template does not differ from the structure of the old template. That is, no columns that have been used in the mapping are in a different position or have a different name. The new *"From"* template can, however, have different selection conditions.

You can also use a *"To"* template as a *"From"* template. However, in this case, the mapping information is not relevant.

Mapping contiguous columns

If you need to map contiguous columns, you might find it more convenient to use the SS, S*n*, or S* commands.

If you select more than one column at a time, ZDT/Db2 displays the From Column Mapping panel for the next selected column each time you press the Exit function key (F3). If you are viewing the From Column Mapping panel for the last selected field and press the Exit function key (F3), Z Data Tools returns you to the Template Mapping panel.

Related tasks

Toggling the selection status of more than one column at a time on page 99

Related references

From Column Mapping panel on page 709
Template Mapping panel on page 900

Column mapping rules

ZDT/Db2 uses the rules shown in Table 2: Column mapping rules on page 119 to determine if a given "From" column can be mapped to a given "To" column. "Yes" indicates that the mapping is valid. "No" indicates that the mapping is not valid.

Note: If mapping, you cannot select a *"From"* column with a data type that is incompatible with the data type of the *"To"* column. For example, Figure 44: From Column Mapping panel showing Delete "From column" selected on page 116 shows the two DATE *"From"* columns, HIREDATE and BIRTHDATE, as un-selectable for mapping to the numeric *"To"* column, BONUS.

Table 2. Column mapping rules

"From" column	"To" column				
	DATE	**TIME**	**TIMESTAMP**	**Character**	**Numeric**
DATE	Yes	No	No	Yes	No
TIME	No	Yes	No	Yes	No
TIMESTAMP	Yes[3]	Yes[4]	Yes[5]	Yes	No
Character[6]	Yes[7]	Yes[7]	Yes[7]	Yes	Yes[8]
Numeric[9]	No	No	No	No	Yes

These rules describe how a validly mapped "From" column is moved to a "To" column. If the "To" column is:

3. Conversion of TIMESTAMP data type to DATE data type results in truncation of time value. Data type conversion is not supported where the Db2® installation-defined format for dates is LOCAL.
4. Conversion of TIMESTAMP data type to TIME data type results in truncation of date and microsecond values. Data type conversion is not supported where the Db2® installation-defined format for time is LOCAL.
5. Conversion between unlike TIMESTAMP data types may result in loss of precision, or padding with zeroes in the fractional part of the second. Similarly the time zone information may be lost, or the default time zone for the current server may be added. For example:

```
TIMESTAMP(4)                    ->      TIMESTAMP(10)
2012-12-02-12.34.56.1234                2012-12-02-12.34.56.1234567890

TIMESTAMP(4) WITH TIME ZONE     ->      TIMESTAMP(0)
2012-12-02-12.34.56.1234-12:00          2012-12-02-12.34.56
```

6. The category "character" includes CHAR and VARCHAR data types.
7. Character value must be in a valid format, depending on the Db2® installation-defined formats for date and time, otherwise the *"To"* column is set to the default value as specified by data create attributes. If the Db2® installation-defined format for dates or time is LOCAL, ZDT/Db2 is unable to validate the *"From"* column before inserting it into the *"To"* column.
8. The "From" column must consist only of numeric characters, and is treated as a numeric column.
9. The category "numeric" includes DECIMAL, NUMERIC, SMALLINT, INTEGER, REAL, DOUBLE, DOUBLE PRECISION, FLOAT, and DECFLOAT data types.

DATE

If the “From” column has a data type of TIMESTAMP, only bytes 1–4 of the “From” column are moved to the “To” column; otherwise, the full four bytes of the “From” column are moved to the “To” column.

TIME

If the “From” column has a data type of TIMESTAMP, only bytes 5–7 of the “From” column are moved to the “To” column; otherwise, the full three bytes of the “From” column are moved to the “To” column.

TIMESTAMP

When the data types of the "FROM" and "TO" columns are identical all bytes of the "From" column are moved to the "To" column.

When the precision and TIME ZONE components of the "FROM" and "TO" columns are different:

- Excess digits are lost in the fraction part of the second when the precision of the "FROM" column is greater than the precision of the "TO" column.
- The fractional part of the second is padded on the right with zeroes when the precision of the "FROM" column is less than the precision of the "TO" column.
- TIME ZONE information is lost when the "FROM" column includes a TIME ZONE and the "TO" column does not.
- TIME ZONE information is added when the "FROM" column does not include a TIME ZONE and the "TO" column does.

Character

The data is aligned at the leftmost character position and, if necessary, truncated or padded with spaces at the right.

If the “From” column is a numeric column, the absolute (unsigned) value is used. If the column is defined as signed and the sign occupies a separate character, that character is not moved, and the sending item is considered to be one less character than the actual size.

Integer or packed

The data is aligned on the assumed decimal point and, if necessary, truncated or padded with zeros. If an assumed decimal point is not explicitly specified in the column definition, one is assumed immediately to the right of the column.

If the “From” column is numeric, the data is moved as if both the “From” column and “To” column were described as signed.

If the “From” column is alphanumeric, the data is moved as if the “From” column were described as a signed numeric.

Internal floating-point

A decimal point is assumed immediately to the left of the column. The data is aligned on the leftmost digit position following the decimal point, with the exponent adjusted accordingly.

External floating-point

The data is aligned on the leftmost digit position, and the exponent adjusted accordingly.

Related tasks

Changing column attributes on page 104

Creating a copybook from a Z Data Tools/Db2 Template

ZDT/Db2 provides a sample REXX exec (HFM2CPYB) that you can use to create a PL/I or COBOL copybook from a Z Data Tools/Db2 template.

To run the exec:

1. Copy HFM2CPYB from the sample library SHFMSAM1 to a REXX library allocated to your TSO session (for example, SYSPROC or SYSEXEC).
2. Issue the statement:

   ```
   HFM2CPYB input_file output_file copybook_type
   ```

   Where:

   input_file

   The name of a Z Data Tools/Db2 template. This is typically a member of a partitioned data set.

   output_file

   The name of an LRECL=80 sequential file, or member of a partitioned data set. The output file contains the copybook derived from the template. Any existing data in the sequential file or member is overwritten.

   copybook_type

   This parameter is optional. It specifies the type of copybook to be produced. Valid values are:

   COBOL

   COBOL copybook created.

   PLI

   PL/I copybook created.

Chapter 5. Viewing and changing Db2® data

To view the contents of a Db2® table or view, use the ZDT/Db2 View Utility (option 1), the ZDT/Db2 Edit Utility (option 2), or the ZDT/Db2 Browse Utility (option B).

The following instructions are given in the context of the ZDT/Db2 editor. In most cases you can perform these actions in View, Edit, or Browse, in which case the term "editor" is used. Similarly, the term "ZDT/Db2 editor session" can indicate either a View editor session, an Edit editor session, or a Browse editor session.

ZDT/Db2 View Utility

The ZDT/Db2 View Utility (option 1) allows you to view and temporarily change data, but without the ability to save any changes.

When you initiate the ZDT/Db2 editor with the View Utility, this is referred to as being in a "View editor session" (or just "View").

The ZDT/Db2 View Utility, similar to ISPF view, provides a Z Data Tools/Db2 editor session that works essentially the same as the Edit Utility. It presents you with an interface that is capable of holding your temporary changes to data. This may be useful, for example, for assessing a change but without any risk of altering live data. The first time you make a change to the data in a View editor session, ZDT/Db2 displays the following warning:

```
Save is not possible in a View session. Use Edit if you want to be able to
save changes.
```

When you exit from a View editor session by pressing the Exit function key (F3), if you have made a change to the data in the ZDT/Db2 editor session, ZDT/Db2 displays a warning as shown in Figure 47: Warning displayed when exiting from a View session after data changes on page 122.

Figure 47. Warning displayed when exiting from a View session after data changes

```
 Process   Options   Utilities   Help
— ┌──────────────────────── View Warning ────────────────────────┐ ──────────────
F │                                                              │   Top of 42
T │          You are currently in Data View mode:                │  Format TABL
  │                                                              │O HIREDATE
  │ Press Enter to confirm exit from View.  No changes will be   │  #7
  │ saved.                                                       │  DATE
  │                                                              │  <---+---->
*  Enter the END or EXIT command to return to View.
0 │                                                              │  01.01.1965
0 │                                                              │  10.10.1973
0 │ Command ===> ____________________________________________    │  05.04.1975
0 │  F1=Help       F2=Split       F3=Exit        F7=Backward    │  17.08.1949
0 │  F8=Forward    F9=Swap       F12=Cancel                     │  14.09.1973
0 └──────────────────────────────────────────────────────────────┘  30.09.1980
000007 000090 EILEEN<      W       HENDERSON<      E11      5498    15.08.1970
000008 000100 THEODORE<    Q       SPENSER<        E21      0972    19.06.1980
000009 000110 VINCENZO<    G       LUCCHESI<       A00      3490    16.05.1958
Command ===>                                                     Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

ZDT/Db2 Edit Utility

The ZDT/Db2 Edit Utility (option 2) allows you to view and change your data (by inserting, deleting and otherwise modifying the contents of a Db2® table or view), and save any changes you make.

When you initiate the ZDT/Db2 editor with the Edit Utility, this is referred to as being in an "Edit editor session" (or just "Edit").

ZDT/Db2 Browse Utility

The ZDT/Db2 Browse Utility[10] (option B) allows you to view data, but without the ability to save any changes.

When you start up the ZDT/Db2 editor in browse mode, this is referred to as being in a "Browse editor session" (or just "Browse").

To start up a Browse editor session, perform one of these actions:

- From the Primary Option Menu panel, enter the `B` command.
- When you are in a Z Data Tools/Db2 editor session, enter the `BROWSE` command.
- From a list of Db2® objects, where it is legitimate to do so, enter the `B` command.

When you start up a Browse editor session, ZDT/Db2 displays the Db2® Browse entry panel.

Related references

Working with lists of Db2 objects on page 303
Table Browse panel on page 866
Table Edit panel on page 868
Table View panel on page 882
Db2 Browse panel on page 612
BROWSE primary command on page 925

ZDT/Db2 editor sessions

This topic tells you how to start and end a Z Data Tools/Db2 editor session, and explains some of the possibilities, such as specifying *"normal mode"* or *"large mode"*, and validating and saving changes.

Starting a Z Data Tools/Db2 editor session

To start a Z Data Tools/Db2 editor session, select one of these options from the Primary Option Menu panel:

1 View

Displays the Db2® View panel.

2 Edit

Displays the Db2® Edit panel.

B Browse

Displays the Db2® Browse panel.

10. For consistency with ISPF, the ZDT/Db2 Browse utility does not appear on the ZDT/Db2 main menu. You can, however, select the ZDT/Db2 Browse utility by typing the B command.

Note: The View and Browse options do not allow you to save any changes you make to the data.

Related references

Primary Option Menu panel on page 760
Db2 View panel on page 631
Db2 Edit panel on page 616

Specifying the editor session mode: "normal mode" or "large mode"

The ZDT/Db2 editor can operate in either "normal mode" or "large mode".

Normal mode

To operate the ZDT/Db2 editor session in "normal mode", enter a value *other than* 0, *, or ALL in the **Row count** field on the editor entry panel.

The ZDT/Db2 editor retrieves rows from the Db2® object and loads them into the editor until one of these conditions occurs:

- The end of the Db2® object is reached.
- The number of rows loaded into the editor equals the value specified in the **Row count** field.
- Available storage is exhausted.

For information about the amount of storage used by the ZDT/Db2 editor to store each row, see Storage considerations on page 138.

For information about limiting the number of rows loaded into an editor session, see Restricting the rows to view by specifying row selection criteria on page 139.

Large mode

To operate the ZDT/Db2 editor session in "large mode", enter `0`, `*`, or `ALL` in the **Row count** field on the editor entry panel. The ZDT/Db2 editor retrieves rows from the Db2® object and loads them into the editor until one of these conditions occurs:

- The end of the Db2® object is reached.
- Approximately three screen's worth of rows has been loaded.

The number of rows displayed on the editor data panel (in tabular format) depends on the terminal emulation settings (24x80, 27x132, 40x132, and so on) and also the number of optional columns headers.

In "large mode", the ZDT/Db2 editor keeps only enough rows in memory to satisfy the current display requirements. You can scroll up or down as required to see all rows of the Db2® object.

In an Edit session in "large mode", the ZDT/Db2 editor also keeps any rows that have been changed in memory, regardless of the current display position. This can increase the storage requirement for the Edit session

substantially if many rows are changed. For more information about limiting the number of rows loaded into an editor session, see Restricting the rows to view by specifying row selection criteria on page 139.

Choosing the appropriate editor mode

The two editor modes, "normal mode" and "large mode", have different characteristics as outlined in Table 3: Characteristics of editor modes on page 125. In most cases, you can use either mode to process a Db2® object.

Table 3. Characteristics of editor modes

	"Normal mode"	"Large mode"
Storage usage	Directly dependent on number of rows loaded, processing objects with large numbers of rows requires very large TSO region size, or not possible.	Independent of number of rows in the Db2® object. Storage use increases as rows in the object are changed.
Db2® requirements	Initial data load occurs quickly for small Db2® objects, more slowly as the size of the object increases. Once loaded scrolling within the Db2® object is very fast.	Initial data load occurs quickly for small Db2® objects, significantly more slowly as the size of the object increases. Once loaded, scrolling to nearby locations is fast, scrolling to distant locations may be slow.
Auto commit feature	Available.	Not available. If selected the auto commit option is disabled. The first attempt to auto commit data changes displays a warning message.
Data visibility	Once loaded, the data in the editor session is static and does not change, even if other processes alter the data.	When an *insensitive cursor* is used, the data displayed in the editor session is static and does not change, even if other processes alter the data. When a *sensitive cursor* is used, the data displayed in the editor session might change if other processes delete or update rows. Rows inserted by other processes are not visible. **Note:** For a description of *insensitive cursor* and *sensitive cursor*, see Editor Options (2 of 8) panel on page 649.
Moving rows[11]	Allowed.	Prohibited.

11. *"Moving a row"* means any operation that changes the *relative* position of a row. These operations are:

- Inserting a new row, or group of rows.
- Repeating a row, or group of rows.
- Moving a row, or group or rows.
- Copying a row, or group of rows to a new position.

All of these operations are prohibited in "large" editor mode.

Table 3. Characteristics of editor modes (continued)

	"Normal mode"	"Large mode"
Sort command	Allowed.	Prohibited.
Row size	No restriction	Row size must not exceed the maximum page size (32704) less 16 bytes. When this limit is exceeded, the editor session reports SQLCODE-670 stating: "The record length of the table exceeds the page size limit". This is because large mode uses a Db2® scrollable cursor, for which each row requires an additional 16 bytes. If you get SQLCODE-670 use "Normal mode" to view and update the Db2® table.

Specifying the start position for your ZDT/Db2 editor session

You can load rows from the Db2® object being processed starting at the first row, or you can start at some other point.

The **Start position** entry field on the editor entry panel specifies the required start position. The default is 1.

When the ZDT/Db2 editor is operating in "normal mode", the editor session contains rows starting at the specified start position. The total number of rows loaded is either the value specified in **Row count**, or some lesser number if the end of the Db2® object was encountered before the required number of rows were loaded. You cannot scroll up to see the rows that were skipped.

When the ZDT/Db2 editor is operating in "large mode", the editor session initially contains rows starting at the specified start position. You can scroll up to see rows prior to the start position, as far as the first row of the Db2® object if required.

Usage Note: You can use the **Start position** value to limit the rows loaded in the editor session, or to start midway through a Db2® object. For example, if you have a large table of, say, a million rows containing people's names ordered by surname and you were only interested in names starting with S, you could enter a start value of, say, 700,000 assuming you knew this is the approximate start position for surnames starting with "S".

Setting options for the current ZDT/Db2 editor session

To set options for the current ZDT/Db2 editor session, display the first of the seven Editor Options panels by entering one of the values shown here against **Edit options** on the Db2® View or Db2® Edit panel:

A "," or an "A"

Displays the first ZDT/Db2 Editor Options panel, prior to displaying the data for the Db2® object being processed. When the first Editor Options panel is displayed, you can navigate to one or more of the other Editor Options panels.

Deleting a row or group of rows is supported in "large" editor mode, since the deleted rows are marked in the editor session with a special marker, and hence the relative position of the rows does not change.

Usage Note: Entering an "A" in an option field changes the behavior of the dialog so that the Editor Options panel is ALWAYS displayed. Use the "/" character when you only need to refer to the editor options on a single occasion. The "/" is cleared after the option is processed returning the dialog to the default behavior, which is to NOT show the various Editor Options panels.

A number in the range 1–8

Displays the *n*th ZDT/Db2 Editor Options panel directly where *n* is the value entered.

Note:
Any options you set on the Editor Options panels apply for the current ZDT/Db2 editor session only. When you exit from the ZDT/Db2 editor session, the editor options revert to the global editor options.

In Edit, an exception to this is the audit option if the installation settings for the currently connected Db2® sub-system specifies AUDIT=OPTIONAL,OFF. In this case there is no difference between the audit option set using the global options, or for an individual Edit session. Further, any value specified is discarded at the end of the ZDT/Db2 session.

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616
Editor Options (1 of 8) panel on page 644

Validating and saving changes with a Z Data Tools/Db2 editor session

In Edit only, you can make changes to the data and save those changes to Db2®.

An Edit session of a Db2® object has important differences when compared to an Edit session of a data set. These differences need to be understood to make the most effective use of a Z Data Tools/Db2 Edit session.

Db2® data is accessed by calls to Db2®. Some important Db2® concepts are discussed here because an understanding of these concepts is central to understanding how the ZDT/Db2 editor operates:

Db2® unit of work

A Db2® unit of work commences when a process initially connects to, or accesses data from, Db2®. The unit of work continues until the process ends the connection or issues an explicit Db2® COMMIT or ROLLBACK statement. Between the start and end of the unit of work, the process may make many calls to Db2® to read (fetch) change (update), insert, or delete rows.

Db2® COMMIT

This is a Db2® command that is used to signal the end of the current unit of work. A Db2® commit occurs automatically when a process ends; however, a Db2® commit may also be issued to finalize any pending changes to Db2® data. Issuing a Db2® commit is an irrevocable step. After the commit, it is not possible to reverse (back out) any changes made since the start of the unit of work.

Db2® ROLLBACK

This is a Db2® command that is used to signal that all pending changes for the current unit of work are to be reversed (backed out). After the rollback is processed, the Db2® data is in the same state that it was in at the start of the unit of work, or at the last commit point.

For more information about Db2® units of work, commit, and rollback, see "Db2® Concepts (Application Processes, concurrency and recovery)" in the *SQL Reference* manual, for the appropriate version of Db2®.

Ordering of rows within a Db2® object is not guaranteed

The relational model on which Db2® is based does not require data within a Db2® table to be in any specific order. This is not to say that data in a Db2® object is not stored in a particular order as this is certainly the case. In addition, a number of methods can be used (for example, CLUSTERING) to influence the way rows are stored in a Db2® table. In terms of the ZDT/Db2 editor, these considerations apply:

- Data is read from Db2® in the order Db2® delivers it to ZDT/Db2. Unless there is an ORDER BY clause for the data entered explicitly (on the Column Selection/Edit panel) or implicitly (for example, when processing a VIEW and the VIEW includes an ORDER BY clause), no assumptions should be made about the relative positions of any two rows.
- Moving rows from one position to another in a Z Data Tools/Db2 Edit session has no effect on the way ZDT/Db2 stores the data in the table.
- When the editor is operating in "large mode", moving rows from one position to another is disabled.

Viewing and changing LOB data and XML documents

The ZDT/Db2 editor handles LOB and XML data differently to the data for other columns, primarily because of the potentially very large size for LOB and XML data.

You can use the editor options (see Editor Options (8 of 8) panel on page 680) to control how much (if any) of the data in LOB or XML columns is shown in the ZDT/Db2 editor. By default the column name for LOB or XML data is shown in the ZDT/Db2 editor, but no data is shown - this minimizes the use of storage by the ZDT/Db2 editor.

When the Db2® object being processed contains LOB columns, you can view and change the data in a single LOB column in a separate ZDT/Db2 editor session, initiated using the LOBBRWS, LOBEDIT or LOBVIEW editor primary commands.

When the Db2® object being processed contains XML columns, you can view and change the data in a single XML column in a separate editor session, initiated using the XMLBRWS, XMLEDIT or XMLVIEW editor primary commands. If the size of the XML document is less than or equal to 10 MB, the ISPF editor is used. Otherwise, the Z Data Tools editor is used.



Note: Do not use the SAVE command in an XMLEDIT or LOBEDIT session because the data will not be updated in the Db2 table.

A Z Data Tools/Db2 editor session of the data in a LOB column has some differences when compared with a Z Data Tools/Db2 editor session of a Db2® object, as described in Table 4: LOB edit differences on page 129. In this table, the term

originating session refers to the ZDT/Db2 editor session of the Db2® object that contains a LOB column; *LOB session* refers to the ZDT/Db2 editor session of a LOB column, this is started from the originating session.

Table 4. LOB edit differences

Action	Object Edit	LOB edit
Initiation	From the function entry panel. EDIT, BROWSE, VIEW editor primary commands	Using the LOBBRWS, LOBEDIT, LOBVIEW editor primary commands from within a Z Data Tools/Db2 editor session of the Db2® object containing the LOB column (Originating session).
Saving	Changes to rows presented to Db2® one at a time. It is possible that Db2® may reject changes to any row for many reasons.	Any changes made in the LOB session are not presented to Db2® as a change to the Db2® object when the LOB session ends. Instead ZDT/Db2 uses a LOB locator to store any changes. When a SAVE command is issued in the originating session, any rows that have pending changes to LOB columns are presented to Db2®.
Commit	A Db2® commit may be issued when a SAVE is issued from the originating session (this depends on the editor option's settings). A Db2® commit is issued at the end of the originating edit session.	No Db2® commits are issued from within a LOB session.
Data Display	Data is shown initially in either TABL or SNGL format, depending on the settings for the 'Initial display format' editor option. (Displayed on the Editor options (1) panel.)	Data is shown initially in either TABL or SNGL format, depending on the 'LOB display format' editor option. (Displayed on the Editor options (8 of 8) panel.) When data is displayed in SNGL mode, you can use the Previous and Next PFKeys (assigned to PF10 and PF11 by default) to scroll through the LOB data.

VARCHAR	The "varying length column" editor options are shown on Editor options (5 of 8) panel. The settings for these options depend on user settings.	The following "varying length column" editor options are fixed:
		`Remove trailing spaces:  Off`
		`Show end of string:  On`
		`Convert spaces to Db2 null:  Off`

Storage	As described for "normal" or "large" mode, see Table 3.	All of the data for the LOB column is loaded into the ZDT/Db2 editor. This may not be possible (insufficient storage) with small TSO region sizes and large LOB columns.

The following usage notes may assist in changing data within a LOB column.

- The LOB session is similar to a Z Data Tools/Db2 editor session of a Db2® object containing a single VARCHAR(*nn*) column, where *nn* is in the range 1000 - 4000. The values for *nn* are approximate and depend on the screen size and other factors.
- The data for a single LOB column is split into pieces as it is loaded in the LOB session. This is purely for convenience, since it makes displaying and changing the LOB data easier, given the limitations of a 3270 terminal.
- You can use the FIND editor command to locate specific text within the LOB session. However a string will not be found if it is split across two "rows" in the LOB session.
- The LOB session initially displays the LOB data in either SNGL or TABL mode, this is determined by the current setting of the 'LOB display format' option. The amount of data shown depends on the 'LOB piece size' option, the terminal's characteristics and and the number of characters in the LOB column's name.
- For large LOBs where the intent is to scroll through the data, consider setting the LOB display option to 'Single', combined with a LOB piece size option of 'Single display size'. This allows the maximum use of the available screen size to display the LOB data and the absolute character positions of data is also displayed. You can scroll through the data using the NEXT and PREVIOUS commands (PF11 and PF10).
- For edit operations, consider setting the LOB display option to 'Table', combined with a LOB piece size option of 'Table display size'. In TABL display mode, all of the data for each "row" is completely visible without the need to scroll to the right.
- You can also use Single display as the starting point for edit operations. When the insert point for edit operations is found, switch to TABL display mode and issue the SP(lit) prefix command against the row of interest. This converts the row to many smaller rows that are completely visible in TABL display format.
- Trailing spaces at the end of a "row" are significant, and you should exercise caution in removing these using the editor. When ZDT/Db2 re-assembles the data for the LOB column, it uses the VARCHAR length for each "row" to determine how much data is added to the LOB column. The editor options for varying length columns (see Editor Options (5 of 8) panel on page 667) are set to prevent accidental removal of trailing spaces from a "row", when the data in that "row" is changed.

Note that the end of the row is shown using the display string delimiter for varying length columns (see Editor Options (5 of 8) panel on page 667). You may explicitly set the end for a row using the input string delimiter (see Editor Options (5 of 8) panel on page 667).

XML differences

ZDT/Db2 uses either the ISPF editor or the Z Data Tools editor to display and process changes to data in an XML column. The ISPF editor is used when the size of the XML document is less than or equal to 10 MB. The Z Data Tools and ISPF editors are designed to handle different types of data:

- The ISPF editor is primarily a text editor, and is well suited to handling XML data.
- The Z Data Tools editor is primarily a data editor, and is well suited to handling data that has a format that can be described using a template.

An XML document cannot be described using a template.

Table 5: XML edit differences on page 131 highlights some important differences between a Z Data Tools/Db2 editor session and an ISPF editor session of an XML document.

Table 5. XML edit differences

Action	Object Edit	XML Edit
Initiation	From the function entry panel. Edit, browse and view editor primary commands.	Using the XMLBRWS, XMLEDIT, XMLVIEW editor primary commands from within a Z Data Tools/Db2 editor session of the Db2® object containing the XML column (Originating session).
Saving	Changes to rows presented to Db2® one at a time. It is possible that Db2® may reject changes to any row for many reasons. When a SAVE command is issued any changes made to XML columns are presented to Db2®.	XML documents are saved when either the XMLEDIT session or the ZDT/Db2 editor session ends.
Editor used	ZDT/Db2 editor	If the size of the XML document is less than or equal to 10 MB, the ISPF editor is used. Otherwise, the Z Data Tools editor is used.
Commit	A Db2® commit may be issued at the end of SAVE command, depending on the settings for various editor options.	No Db2® commits are issued from within an edit session of an XML document.
Data Display	Data is shown initially in either TABL or SNGL format, depending on the settings for the 'Initial display format' editor option. (Displayed on the Editor options (1) panel.)	There is only one display format in the ISPF editor. The data is formatted using tag and end-tag pairs prior to display.
Storage	As described for "normal" or "large" mode, see Table 3: Characteristics of editor modes on page 125.	All of the data for the XML document is written to a temporary data set, and then processed using either the ISPF editor or the Z Data Tools editor. The following constraints might limit the size of the XML document that can be processed: • The user's TSO region size. • Allocation restrictions on the size of the temporary data set. • Disk space constraints. • The Db2® LOBVALA parameter (specified in Db2® job DSNTIJUZ). The value of this

Table 5. XML edit differences (continued)

Action	Object Edit	XML Edit
		parameter limits the maximum amount of storage, in KB, that each user is to have for storing LOB values.

Db2-related considerations when changing data in a Z Data Tools/Db2 Edit session

A Z Data Tools/Db2 Edit session is a Db2® unit of work, commencing at the start of the Edit session and continuing until one of these events occurs:

- The Edit session ends when END is pressed.
- The Edit session ends when CANCEL is pressed.
- A Db2® commit point is issued as part of a Z Data Tools/Db2 SAVE primary command.
- A Db2® commit point is issued automatically because the number of successful changes reached the **Auto-commit count** value specified on the Editor Options (6 of 8) panel.

Within a Z Data Tools/Db2 Edit session, multiple changes can be made to the data; you can send these changes to Db2® for validation with the SAVE primary command. At any time, the changes can be backed out (rolled back) to the state the data was in at the start of the Edit session, as long as a Db2® commit point has not been issued. This is unlike an the edit of a data set where typically, once changes have been saved, it is not possible to return to an earlier version of the data.

Changes made during a Z Data Tools/Db2 Edit session are processed one row at a time. This is unlike the edit of a data set where, typically all of the changes made are written all together to the data set when the data is saved and the existing data set is completely replaced by the new one.

Changes made to the data in a Db2® object are validated by Db2® prior to the changes being accepted by Db2®. This is unlike changes that can be made to a data set, where typically there is no validation of the data and issuing a SAVE or END primary command is certain to save any pending changes.

Db2® might reject changes for many reasons. These are some, but not all, of the reasons that a change may be rejected:

- Inappropriate values for the data type of a column. For example, entering non-numeric data in a numeric column, or numeric data that is out of range in a numeric column.
- Violation of any check constraints applying to the Db2® object. For example, a column defined as CHAR(1) used to store the gender of a person, constrained to accept a value of only "M" or "F".
- Violation of any referential integrity rules applying to the Db2® object. For example, attempting to delete a primary key for which there are dependent rows in another table, and the DELETE rule for the constraint is RESTRICT
- Attempts to insert a duplicate with respect to a unique index defined on the Db2® object.

Within a Z Data Tools/Db2 Edit session, you can use these methods to validate, and optionally commit, pending changes to Db2®:

- Issue an explicit SAVE primary command.
- Press the Enter key with no other pending changes (the auto save feature). (See the **Save data** option on Editor Options (6 of 8) panel on page 672.)
- Press the END key (normally PF3) to end the Edit session. (See Ending a Z Data Tools/Db2 editor session on page 133.)

When you issue a SAVE primary command, any pending changes made are processed one row at a time. Each change results in a single INSERT, UPDATE, or DELETE SQL statement. If the change is unsuccessful, ZDT/Db2 marks the row in error in the Edit session. You must then correct the error before you can complete (commit) the data changes to Db2®.

ZDT/Db2 may issue Db2® commit points automatically during the SAVE process, if a non-zero value has been specified for the **Auto-commit count**, specified on the Editor Options (6 of 8) panel. Such commit points are issued whenever there have been the specified number of successful changes; it is possible for multiple commit points to be issued during the SAVE process. The commit points are issued regardless of whether errors are encountered when processing other rows.

At the end of the SAVE process, ZDT/Db2 may also issue an explicit Db2® commit, depending on the setting of the **Commit when save issued** and **Commit when no save errors** options. (See Editor Options (6 of 8) panel on page 672.)

Usage note: When processing a Db2® object that is subject to constraints, frequent validation of any changes made can avoid frustration at the end of the editor session (when PF3 is pressed). You can issue the SAVE command repeatedly during an Edit session to validate the changes just made, while retaining the ability to cancel the entire Edit session. Remember that, until a Db2® commit point is issued, the entire unit of work can be rolled back. In "normal mode", you also have the option to auto-commit changes when you issue a SAVE primary command. This option is disabled in "large mode".

Related references

SAVE primary command on page 999
Editor Options (6 of 8) panel on page 672

Ending a Z Data Tools/Db2 editor session

In View or Browse:

To end your ZDT/Db2 editor session:

Press the Exit function key (F3) or the Cancel function key (F12).

In Edit:

To end your ZDT/Db2 editor session and save any changes you have made:

Enter `END` on the command line, or press the Exit function key (F3).

To present the changes you have made to the current table to Db2® for validation, without ending the ZDT/Db2 editor session:

Use the SAVE primary command.

When you use the SAVE command, ZDT/Db2 processes any changes made one line at a time. Each change results in either an SQL INSERT, UPDATE or DELETE operation, Db2® may reject one of more of your changes. This is unlike an editor session of a normal data set, where generally all changes are made, or no changes are made.

Note the following about the SAVE process:

- ZDT/Db2 issues Db2® COMMIT points according to these commit options:
  - **Commit when save issued**
  - **Commit when no save errors**
  - **Auto-commit count**

  The settings for these options are either the global settings, or as specified for this ZDT/Db2 editor session only. The first two options influence whether a Db2® COMMIT is issued at the end of the SAVE process. The third option influences whether Db2® COMMIT points are issued during the SAVE process.
- Issuing a Db2® COMMIT finalizes all pending changes to the data, and is an irrevocable process. After the COMMIT is issued, you can cancel the editor session; however, any changes made since the start of the edit session are *not* backed out. Conversely, if no COMMIT has been issued, you can back out *all* changes made during the current editor session, regardless of how many SAVE commands have been issued.
- When the editor is operating in "large mode", any automatic Db2® COMMIT is disabled. Your changes are still presented to Db2® and validated; however, the final COMMIT is deferred until you end the editor session.

To end your ZDT/Db2 editor session without saving your changes (that is, since the previous successful commit point):

Enter `CANCEL` (or `CAN`) on the command line, or press the Cancel function key (F12). If you have made any changes to the data, ZDT/Db2 displays a message box asking you to confirm that you want to discard the changes.

Note:

1. If there are dependent related ZDT/Db2 editor sessions still active:
   - You need to end these sessions before you end the current (parent) ZDT/Db2 editor session (unless you cancel the session).
   - Canceling the current ZDT/Db2 editor session also cancels all the dependent related ZDT/Db2 editor sessions.

Related tasks

Editing related tables on page 192
Specifying the editor session mode: "normal mode" or "large mode" on page 124

Related references

- CANCEL primary command on page 926
- SAVE primary command on page 999
- Editor Options (6 of 8) panel on page 672

Starting another ZDT/Db2 editor session

When you are in a Z Data Tools/Db2 editor session, you can start another ZDT/Db2 editor session by issuing the BROWSE, EDIT, VIEW, or REDIT primary commands.

BROWSE, EDIT, VIEW commands

The BROWSE, EDIT, and VIEW primary commands start another ZDT/Db2 editor session and display the Browse, Edit, or View entry panel respectively.

The new ZDT/Db2 editor session is stacked on top of the current ZDT/Db2 editor session. If you specify a Db2® object owner and Db2® object name, the owner and name fields are pre-filled on the entry panel for the new ZDT/Db2 editor session.

Note: To start a Z Data Tools/Db2 editor session in Browse mode from the Primary Option Menu panel, enter the abbreviated form of the BROWSE command, `B`, on the command line.

Examples:

`VIEW`

Starts a new ZDT/Db2 editor session, in View mode.

`BRO DSN8910.EMP`

Starts a new ZDT/Db2 editor session, in Browse mode. The owner field is set to 'DSN8910' and the name field is set to 'EMP', on the Browse entry panel for the new editor session.

`EDIT EMP`

Starts a new ZDT/Db2 editor session, in Edit mode. The name field is set to 'EMP', on the Edit entry panel for the new editor session.

To exit the new ZDT/Db2 editor session and return to the current ZDT/Db2 editor session, press the End or CANCEL key.

REDIT command

The REDIT primary command displays information about tables related by a Db2® referential integrity (RI) constraint, to the Db2® object currently being edited. The command is context and cursor sensitive.

Related references

- Listing related tables on page 193
- BROWSE primary command on page 925

EDIT primary command on page 940
VIEW primary command on page 1010
REDIT primary command on page 992
Related tables panel on page 787

Specifying the Db2® object you want to view

This section describes how you specify the Db2® object you want to view when you use the ZDT/Db2 View Utility (option 1), the ZDT/Db2 Edit Utility (option 2), or the ZDT/Db2 Browse Utility (option B) to view Db2® data or, in the case of edit, to optionally change Db2® data. The descriptions apply to the Db2® View panel, Db2® Edit panel, and Db2® Browse panel respectively.

To specify the table you want to view, use the **Name** entry field and, optionally, the **Location**, **Owner**, **Database**, and **Table space** entry fields.

To specify the template that you want ZDT/Db2 to use when viewing the data, use a combination of the **Template** and **Template usage** entry fields. You can:

- Specify a template that describes the format of the Db2® object:
  - Use **Data set name**, and optionally **Member**, to specify the template.
  - Select 1 (Above).
- Use the last (previously used) template:
  - Select 2 (Previous).

  Any values in the Data set name and Member fields are ignored.
- Use ZDT/Db2 to generate a template using the information in the Db2® catalog:
  - Select 3 (Generate from table). This is the default setting.

  Any values in the Data set name and Member fields are ignored.
- Use ZDT/Db2 to generate a template using the information in the Db2® catalog and store the generated template using the name you specify (if the data set already exists, ZDT/Db2 replaces it):
  - Use **Data set name**, and optionally **Member**, to specify the name that you want ZDT/Db2 to use if storing the generated template.
  - Select 4 (Generate/Replace).

You can select this option to change the options for the current editor session:

Edit options

Enter one of these:

A "/" or an "A"

Displays the first ZDT/Db2 editor options panel, prior to displaying the data for the Db2® object being processed.

A number in the range 1–8

Displays the *n*th ZDT/Db2 editor options panel directly where *n* is the value entered.

Any options that you set last only for the duration of the current Edit session, and override any conflicting global options. To permanently change the Edit options, select option 0.2.

You can select the following options by entering a "/" or an "A":

Edit template

Select this option to edit the template before editing the data.

Re-edit template

Select this option to control the navigation between panels within the Edit dialog. This option controls the behavior of the F3 and F12 keys when pressed from within an Edit session.

Create audit trail

Select this option to generate an audit trail report of all successful modifications to Db2® data made during the current Edit session.

Note: This option is not displayed if SAF-rule controlled auditing is not in effect. See SAF-rule controlled auditing on page 71.

Note: Entering an "A" in an option field changes the behavior of the dialog so that the appropriate options panel is *always* displayed. If you only need to refer to an options panel on one occasion, use the "/" character. The "/" is cleared after the option is processed, thereby returning the dialog to the default behavior, which is to *not* show the various option panels.

Related tasks

Selecting options on ZDT/Db2 panels on page 45
Setting options for the current ZDT/Db2 editor session on page 126
Editing a template on page 84

Related references

Db2 Browse panel on page 612
Db2 Edit panel on page 616

Restricting the data that ZDT/Db2 loads from Db2®

When you view a Db2® table or view in a Z Data Tools/Db2 editor session, ZDT/Db2 accesses Db2® to obtain a copy of the data in the table or view. This copy of the data is loaded into the ZDT/Db2 editor. In normal circumstances, the data for every column of the Db2® object is retrieved - with the exception of any LOB or XML columns (see Viewing and changing LOB data and XML documents on page 128 for more information). This occurs regardless of whether you have deselected one or more columns using the template editor.

However, you may find that you want to restrict the data that ZDT/Db2 loads, either because you are only interested in data that satisfies certain selection criteria, or simply to limit the number of rows of data that ZDT/Db2 retrieves.

The following sections discuss the ways in which you can restrict the data that ZDT/Db2 loads.

Related tasks

Storage considerations on page 138
Viewing large tables or views on page 139
Restricting the rows to view by specifying row selection criteria on page 139
Restricting the rows to view with the "Row count" entry field on page 140
Interactively developing row selection criteria on page 139
Using Select Statement Edit to limit the data retrieved by ZDT/Db2 on page 140
Specifying the editor session mode: "normal mode" or "large mode" on page 124
Specifying the start position for your ZDT/Db2 editor session on page 126

Storage considerations

If you are working with large Db2® tables or views, you may need to consider how much space is required in the editor to store the copy of the Db2® object.

The ZDT/Db2 editor uses storage in the TSO user's address space. The size of the address space is governed by the TSO region size, which may be limited by your systems administrator. As a guide, the amount of space needed by the ZDT/Db2 editor is the number of rows loaded, multiplied by the number of bytes of data in each row, multiplied by an editor overhead.

For example, to load 100,000 rows of 20 bytes into the ZDT/Db2 editor, requires at least 2MB of storage. The editor overhead factor is at least 1 and can be as much as 3, so up to 6MB of storage may be needed.

To take another example, if each row occupies 1000 bytes, every 1024 rows loaded requires about 1MB of storage. Therefore to load 100,000 such rows, at least 100MB of storage is required.

When the editor is operating in "normal mode", the number of rows loaded remains fixed and can be used to estimate the total storage requirements.

When the editor is operating in "large mode", the number of rows loaded in approximately three times the number of rows visible in TABL display format. This number remains unchanged regardless of any scrolling operations. When changes

are made to rows, those rows remain in the editor. Therefore you may still encounter storage problems if you perform edit operations that change large numbers of rows in the table.

When you need to change large numbers of rows, consider changing the table by direct execution of SQL statements.

Viewing large tables or views

If the Db2® table or view requires more storage than is available in the TSO region, you will *have* to restrict the data that ZDT/Db2 loads from Db2®. The most convenient way to do this is to select "large mode" by specifying `ALL`, `0` or `*` in the **Row count** field on the entry panel. Large mode may not be available if the product installer has chosen to disable it.

When processing an object containing large object columns, ZDT/Db2 requires at least as much memory (virtual storage) as that needed to store one row of data from the object, with every large object column at the maximum possible size for that column. This may not be possible with a small TSO region size and large object data in the megabyte (or greater) range. To process an object with large object columns, create a Db2® view of the object without the large object columns present, and then process the view.

The following sections describe other ways in which you can restrict the data that ZDT/Db2 loads.

Related references

Specifying the editor session mode: "normal mode" or "large mode" on page 124

Restricting the rows to view by specifying row selection criteria

When you specify row selection criteria, ZDT/Db2 retrieves only those rows that match your criteria from Db2®. Only these rows are loaded into the Db2® editor. Specifying row selection criteria is usually the most convenient and efficient method for restricting the data that ZDT/Db2 loads from Db2®.

This method applies to both "normal mode" and "large mode".

Related tasks

Selecting rows on page 86
Specifying the editor session mode: "normal mode" or "large mode" on page 124

Interactively developing row selection criteria

Sometimes you may need to view or edit a Db2® table or view with which you are not familiar. You may only need to view (and possibly change) a few rows of the table or view. ZDT/Db2 View and ZDT/Db2 Edit provide options to assist in interactively identifying the rows of interest, and then restricting the data loaded by ZDT/Db2 using row selection criteria.

To use this method, you need to alter the behavior of the editor dialog by setting the **Edit template** and **Re-edit template** options (use *"A"*). You can then alternate between the data display and the Column Selection/Edit panels, refining the row selection criteria until you have identified exactly the rows you require.

Related tasks

Selecting options on ZDT/Db2 panels on page 45
Selecting rows on page 86

Restricting the rows to view with the "Row count" entry field

If you just want to limit the volume of data that ZDT/Db2 retrieves from Db2®, use the **Row count** entry field on the Db2® View panel or the Db2® Edit panel to specify the number of rows of Db2® data you want.

You can use this entry field to select "large mode" by typing `ALL`, `0`, or `*` in the field.

If the maximum number of rows was retrieved, the message *"Limit of Rows Retrieved"* appears immediately below the last row (in TABL display format).

You can use this entry field when viewing or editing large tables, in combination with the **Start position** entry field, to show small portions of the table. You can then specify row selection criteria that limit rows retrieved to those of interest.

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616

Restricting the rows to view with the "Start position" entry field

In some situations, the rows of interest may be located together in the table, but not necessarily at the beginning. In this case, if you know the approximate starting row number, you can specify a start position value in the **Start position** entry field. ZDT/Db2 skips rows until the specified start position and then loads rows from that point.

In "normal mode", the **Row count** value applies from the starting row, not the beginning of the table. Once the rows have been loaded you cannot scroll up to see the skipped rows.

In "large mode", ZDT/Db2 loads rows from the start position until the limits discussed in Specifying the editor session mode: "normal mode" or "large mode" on page 124 are reached. In "large mode", you can scroll up from the start position to the first row of the table if required.

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616

Using Select Statement Edit to limit the data retrieved by ZDT/Db2

You can use the *"Enter, execute and explain SQL statements"* function (option 4.3) to enter a SELECT statement and display the results in a Z Data Tools/Db2 editor session. In general, if you want to view or edit a Db2® table or view, View (option 1) and Edit (option 2) are recommended. However, you can specify the columns to retrieve using option 4.3.

For example, the following statement retrieves all columns from the DSN8810.EMP table:

```
SELECT * FROM DSN8810.EMP
```

However, the following statement retrieves only the listed columns:

```
SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO,
HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE FROM DSN8810.EMP
```

You might use this method when editing a table with many columns, most of which are of no interest to you. Alternatively, if the Db2® object has a *"wide"* row, omitting long VARCHAR or VARGRAPHIC columns may dramatically reduce the amount of data retrieved by ZDT/Db2 and loaded into the editor.

Note: When you omit columns from a Db2® object, you may find that not all edit operations are possible. For example, inserting a new row will fail if any of the omitted columns is defined as NOT NULL.

Related tasks

View and Edit options (options 1 and 2) on page 67
Storage considerations on page 138
Entering, executing, and explaining SQL statements on page 424

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616
Editor Options (1 of 8) panel on page 644

Viewing data in a Z Data Tools/Db2 editor session

When you view a Db2® table or view in a Z Data Tools/Db2 editor session, you can:

- Locate a specific row or column
- Search the data for a specific string or numeric value
- Select a display format for the data
- Scroll or sort the data

Count of rows fetched in a Z Data Tools/Db2 editor session

When ZDT/Db2 first displays data in an editor session, it shows the number of rows that were fetched. *"`nn rows fetched`"* is shown in the upper left, on either the second line (if the command line is at the bottom of the display) or third line (if the command line is at the top of the display), and persists until you press Enter or any function key.

Locating a row

To move to a particular row number, use the LOCATE primary command. In a Z Data Tools/Db2 editor session, rows are numbered from 1 upwards and the row number is shown in the prefix area (TABL display mode) or in the upper right of the panel (SNGL display mode). For example, to move to row 42, enter the following command:

```
LOCATE 42
```

In TABL display format, this command scrolls the display (either forwards or backwards) so that row number 42 is displayed at the top of the screen. In SNGL display format, the display changes so that row number 42 is displayed.

In a Z Data Tools/Db2 editor session, you can also use the LOCATE primary command to move to a row to which a particular label has been assigned. For example, to move to the row labeled *".HERE"*, enter the following command:

```
LOCATE .HERE
```

In a Z Data Tools/Db2 editor session, if in TABL display format, this command scrolls the display (either forwards or backwards) so that the row labeled *".HERE"* is at the top of the screen. In SNGL display format, the display changes so that the row, previously labeled *".HERE"* in TABL display format, is displayed.

Related tasks

TABL display format on page 153
SNGL display format on page 156
Assigning labels to rows on page 150

Related references

LOCATE primary command on page 978

Locating a column

You can use the LOCATE primary command to move to a particular column. You can specify either a Db2® column number, or part or all of a column name.

You specify a Db2® column *number* by preceding the column number with the *"#"* symbol (for example, #4). Db2® column numbers are shown on the Column Selection/Edit panel, and optionally in both TABL and SNGL display formats when you view data. (To display column numbers, you must select the **Show column number** option on the third Editor Options panel.) For long character columns only (> 50 bytes) you can optionally locate a character position within a column.

You specify a Db2® column *name* by typing all or part of the column name, with or without quotation marks.

When ZDT/Db2 finds the specified column, it scrolls the data so that the matching column is displayed on the left of the screen (in TABL display format), or at the top of the screen (in SNGL display format). In addition, after ZDT/Db2 has found the specified column, in SNGL display format if you press F11 (to display the following record) or F10 (to display the

preceding record), the data is scrolled to show the matching column at the top of the screen for each record displayed. If you change to TABL display format, the data is scrolled so that the matching column is displayed on the left of the screen.

Related tasks

TABL display format on page 153
SNGL display format on page 156
View and Edit options (options 1 and 2) on page 67
Setting options for the current ZDT/Db2 editor session on page 126

Related references

LOCATE primary command on page 978
Column Selection/Edit panel on page 497
Editor Options (3 of 8) panel on page 655

Specifying the column number or column name

To locate a column, you must specify one of the following:

- a Db2® column number
- a column name, or
- part of a column name

The following command locates column 4 in the data:

```
LOCATE #4
```

The following command locates column 4 in the data, and scrolls the display so that character position 100 of that column is visible:

```
LOCATE #4 100
```

Either of the following commands locates column EMPNO in the data:

```
L EMPNO
```

```
L 'EMPNO'
```

If you specify part of column name, the LOCATE command matches any column that contains the specified string within the column name. For example,

```
L NO
```

locates both EMPNO and DEPTNO.

Using the FLD parameter

The FLD parameter indicates to ZDT/Db2 that the string following the FLD keyword is the name or part of the name of the column you want to locate. You use the FLD parameter to resolve any ambiguity that might exist between column names and other command parameters.

You *must* specify the FLD parameter before the string, if the search string is *"NEXT"*, *"PREV"*, *"FIRST"*, *"LAST"*, *"FLD"*, starts with *"#"*, or is a number.

For example, to locate a column name containing the # symbol such as #ITEMS, use the following command:

```
L FLD #ITEMS
```

The command `L #ITEMS` results in an error as ZDT/Db2 is expecting the # symbol to be followed by a numeric Db2® column number.

Specifying the search direction and starting point

By default, the LOCATE primary command searches for the next occurrence of the specified column name, starting at the column currently displayed on the left of the screen (in TABL display format), or at the top of the screen (in SNGL display format).

You can control the direction and scope of the column name search by entering one of the following optional parameters:

- NEXT
- PREV
- FIRST
- LAST

The following sections describe how you use each of the preceding parameters.

Using the NEXT parameter

NEXT is the default value. Both of the following commands search for the next column name containing the string *"NO"*:

```
L NEXT 'NO'
L NO NEXT
```

In TABL display format, the preceding commands start at (but exclude) the column currently displayed on the left of the screen and scan right to find the first column name that is either *"NO"*, or contains the string *"NO"* (for example, NO, EMPNO, DEPTNO and NOMINAL all contain the string *"NO"*). The search ends when either a match occurs, or the last column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed on the left of the display.

In SNGL display format, the preceding commands start at (but exclude) the column currently displayed at the top of the screen and scan down to find the first matching column name. The search ends when either a match occurs, or the last column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed at the top of the display.

Using the PREV parameter

Both of the following commands search backwards for the next column name containing the string *"NO"*:

```
L PREV NO
L NO PREV
```

In TABL display format, the preceding commands start at (but exclude) the column currently displayed on the left of the screen and scan left to find the first column name that is either *"NO"*, or contains the string *"NO"*. The search ends when either a match occurs, or the first column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed on the left of the display.

In SNGL display format, the preceding commands start at (but exclude) the column currently displayed at the top of the screen and scan up to find the first matching column name. The search ends when either a match occurs, or the first column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed at the top of the display.

Using the FIRST parameter

Both of the following commands search for the first column name containing the string *"NO"*:

```
L FIRST NO
L 'NO' FIRST
```

In TABL display format, the preceding commands start at the first column for the Db2® object and scan right to find the first column name that is either *"NO"*, or contains the string *"NO"*. The search ends when either a match occurs, or the last column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed on the left of the display.

In SNGL display format, the preceding commands start at the first column for the Db2® object and scan down to find the first matching column name. The search ends when either a match occurs, or the last column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed at the top of the display.

Using the LAST parameter

Both of the following commands search for the last column name containing the string *"NO"*:

```
L LAST NO
L NO LAST
```

In TABL display format, the preceding commands start at the last column for the Db2® object and scan left to find the last column name that is either *"NO"*, or contains the string *"NO"*. The search ends when either a match occurs, or the first column of the Db2® object is encountered. If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed on the left of the display.

In SNGL display format, the preceding commands start at the last column for the Db2® object and scan up to find the first matching column name. The search ends when either a match occurs, or the first column of the Db2® object is encountered.

If a matching column name is found, ZDT/Db2 scrolls the data so that the matching column is displayed at the top of the display.

Finding a string

To search the data for a specific string, use the FIND primary command (which you can abbreviate as F or /).

For example, to search for the next occurrence of the string *"Golden"* in any mix of uppercase and lowercase, enter the following on the command line:

```
F GOLDEN
```

Repeating the search

To move the cursor to the next occurrence of the search string, issue the RFIND primary command or press the RFind function key (F5).

Note: RFIND works from the cursor position. When you use the RFind function key (F5), the cursor remains in the body of the data and pressing the key moves to the next occurrence of the string. However, if you type the RFIND primary command on the command line, you need to reposition the cursor on the last occurrence of *string* and then press Enter in order to find the next occurrence.

To find the same string as you specified on the previous FIND primary command, specify an asterisk (*) as the search string.

Handling long strings

For character columns (CHAR, VARCHAR), you can specify a search string up to 100 characters long. However, when you are dealing with a long search string, you may find that you cannot fit the whole FIND command (including other parameters) on the command line. To overcome the problem, do one of the following to display an Extended Command Entry pop-up panel:

- Enter `FX` (for Find Extended)
- Enter the FIND command (or one of its abbreviations, such as F or /) *with no parameters*

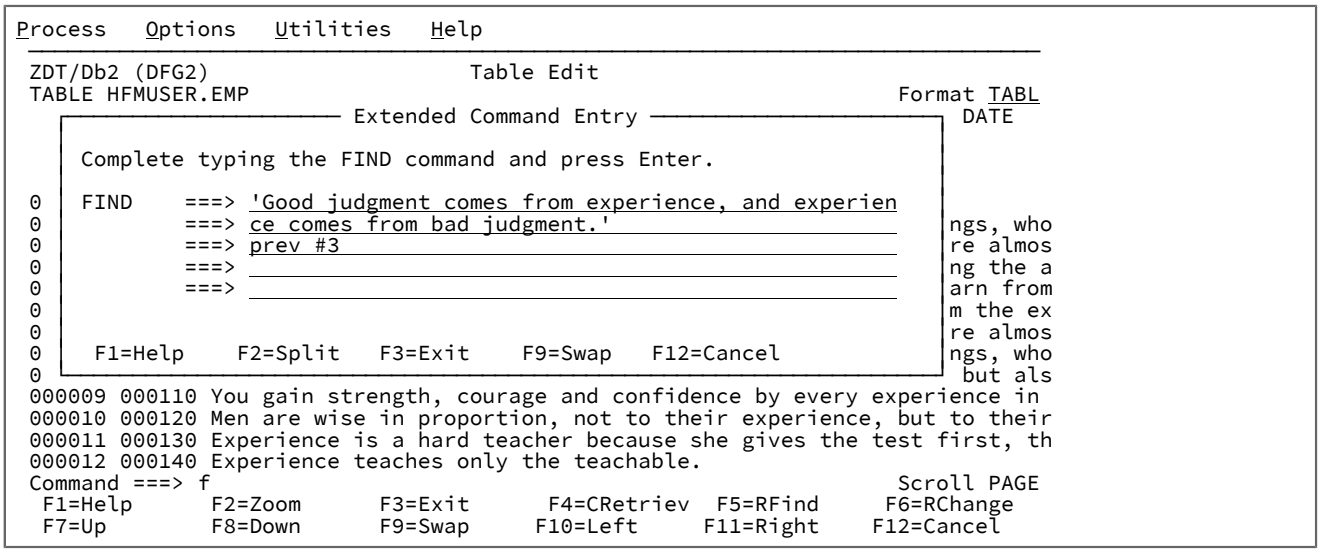
Note: If you have previously used the FIND or CHANGE commands in the current ZDT/Db2 editor session, to display the Extended Command Entry pop-up panel in this way, you will need to first use the RESET command before entering the FIND command (or one of its abbreviations) without any parameters. Alternatively, enter `FX`.

The Extended Command Entry pop-up panel contains five lines (each 50 characters long) that you can use to enter all the FIND command parameters (but not the actual keyword FIND or any of its abbreviations). ZDT/Db2 treats the five lines on the pop-up panel as contiguous, allowing you to specify a long search string, the Db2® column number and optionally any other parameters, using up to 250 characters in total.

Note: You can continue the search string (or any other parameter) from the last position on one line of the pop-up panel to the first position of the next line. However, this does not mean that you need to completely fill a line before using the next line. For example, you can specify the search string on the first of the five lines, and the Db2® column number (specified as #*n*) on the next line. If you do not use all of a line, ZDT/Db2 treats the unused part of the line as a single space.

Figure 48: Extended Command Entry pop-up panel for the FIND command showing contiguous nature of entry lines on page 147 shows the Extended Command Entry pop-up panel for the FIND command. In this example, ZDT/Db2 finds any previous occurrences of a long string (78 characters) in Db2® column 3.

Figure 48. Extended Command Entry pop-up panel for the FIND command showing contiguous nature of entry lines



Related references

FIND primary command on page 953

Finding data within specific columns

You can optionally limit the effects of primary commands to a specified column, a list of columns, or one or more ranges of columns.

You specify the columns to be searched using #*n* column references. These are displayed above each column in TABL display format, or next to each column in SNGL display format. The search string is found only if it is completely contained within one of the specified columns.

Specifying a list of columns

When you specify a list of columns, you must either:

- Enclose the column references (optionally separated by commas) in parentheses, or
- Separate the column references with commas but without any intervening spaces.

These are valid lists of columns:

```
#17
(#17)
(#17 #22)
(#17, #22)
#17,#22
```

These are invalid lists of columns:

```
#17 #22
#17, #22
```

The search string is found if it is completely contained within one of the specified columns. For example, the following command excludes all records containing the characters "the" (in any mix of uppercase and lowercase) in column #5:

```
EXCLUDE ALL the #5
```

The following command finds the next record containing an uppercase letter A in column #8 or #9:

```
FIND C'A' #8,#9
```

Specifying ranges of columns

You specify a range of columns by nominating the first and last column reference of the range, separated by a hyphen, with or without intervening spaces. You can optionally enclose the range in parentheses.

These are valid ranges of columns:

```
#17-#19
#17 - #19
(#17-#19)
(#17 - #19)
```

To specify more than one range of columns:

- Enclose each column range in parentheses (optionally separated by commas), or
- Separate each column range with commas, but without any intervening spaces.

These are valid ranges of columns:

```
(#8-#11) (#17-#19) (#24-#25)
(#8-#11),(#17-#19),(#24-#25)
#8-#11,#17-#19,#24-#25
```

These are invalid ranges of columns:

```
#8-#11 #17-#19 #24-#25
#8-#11, #17-#19, #24-#25
```

Note:

1. Column ranges can overlap. For example, `#2-#7, #5-#9`. In this case, the result is a consolidated single column range equivalent to `#2-#9`.

TABL display format on page 153
SNGL display format on page 156
Searching numeric columns on page 152
Excluding rows on page 162

Related references

CHANGE primary command on page 928
EXCLUDE primary command on page 943
FIND primary command on page 953

Finding a string within specific portions of the data

When you use the FIND command to search for a string, you can limit the search to specific portions of the data you are searching:

- To limit the search for a string to only where it appears as a *prefix* in the data, specify the PREFIX parameter.
- To limit the search for a string to only where it appears as a *suffix* in the data, specify the SUFFIX parameter.
- To limit the search for a string to only where it appears as a *word* in the data, specify the WORD parameter.

Related references

FIND primary command on page 953

Finding data within excluded or non-excluded rows

If your data contains excluded and non-excluded rows, you can limit the search for a string or numeric value to:

- Only non-excluded rows (parameter NX), or
- Only excluded rows (parameter EX or X)

For example, the following command finds the next non-excluded row containing the string *"SCARECROW"* in column number 1 or column number 7:

```
FIND SCARECROW (#1 #7) NX
```

And the following command finds the next excluded row containing the value 1024 in column number 3:

```
FIND 1024 #3 EX
```

Related references

FIND primary command on page 953

Finding data within a label range

In a Z Data Tools/Db2 editor session, you can limit the search for a string or numeric value to a range of rows, identified by a pair of labels indicating the first and last rows to be searched.

For example, the following command finds the next row containing the string *"TIME"* in column number 10 in the range of rows from the label .STRT to the label .FNSH

```
FIND TIME #10 .STRT .FNSH
```

And the following command finds the next row containing the value 2840 in column number 4 or column number 6 in the range of rows from the current cursor position to the label .END

```
FIND 2840 #4,#6 .ZCSR .END
```

Related tasks

Assigning labels to rows on page 150

Related references

FIND primary command on page 953

Assigning labels to rows

In a Z Data Tools/Db2 editor session, if you use the CHANGE, DELETE, EXCLUDE, EXPORT, or FIND command in TABL display format, you can limit the rows that are searched by entering a *"range"* parameter. The range parameter consists of a pair of *labels* indicating the first and last rows to be searched. The string is found if it is contained within the specified range.

A label can be one to four alphabetic characters, beginning with a period (.). Labels starting with the letter "Z" are assigned by the editor.

You assign a label by typing the valid character string into the prefix area of the row. In the following example, the label .HERE is assigned to row number 16 and the label .THRE is assigned to row number 129:

```
⋮
00015 This is a row.
.HERE This is a row with a label.
00017 This is a row.
⋮
00128 This is a row.
.THRE This is a row with a label.
00130 This is a row.
⋮
```

After it is assigned, a label stays with the row, even if the row is moved.

To remove a label from a data row, blank out the label characters (or overtype them with a new label) then press Enter. You can unassign a label by deleting the row containing that label, or you can remove all row labels by using the RESET LABEL command.

You can overtype a label with a prefix command without unassigning the label. The label reappears at the completion of the command and remains displayed in the row prefix area.

You can move a label to another row by typing the same label on a new row.

Some labels are automatically assigned and maintained by the editor. These special labels begin with the letter “Z” (Your own labels therefore must not start with a “Z”.) Unlike other labels, these editor-assigned labels do not necessarily stay with the same row, but instead represent logical positions on the display.

Some of these editor-assigned labels are available to you. The editor assigns:

.ZCSR

To the row on which the cursor is currently positioned.

.ZFST

To the first row (relative row number 1). This can be abbreviated to .ZF.

.ZLST

To the last row. This can be abbreviated to .ZL.

The following FIND command finds the next row, excluded or non-excluded, containing the string *xxx*, in the column with column number 5, in the range of rows from the current cursor position to the end of data:

```
FIND xxx #5 .ZCSR .ZLST
```

The following EXCLUDE command excludes all rows, excluded or non-excluded, containing the string “ABC”, in either of the columns with column numbers 3 and 7, in the range of rows from the label .HERE to the label .THRE:

```
EXCLUDE ALL ABC (#3 #7) .HERE .THRE
```

If you use the range parameter, the two labels must be separated by a blank or comma. A single label is not allowed. The labels can be any combination of the editor-assigned labels (.ZCSR, .ZFST, .ZLST) or labels you have assigned yourself.

The label representing the smaller relative row number is used as the start of the range and the label representing the larger relative row number is used as the end of the range, regardless of the order in which they are specified.

If the range you specify includes excluded rows that are hidden from display or represented by shadow lines, those rows are not affected by the command.

Related references

CHANGE primary command on page 928
DELETE primary command on page 936
EXCLUDE primary command on page 943

FIND primary command on page 953
RESET primary command on page 994

Searching numeric columns

The numeric value of the search string must equal the formatted numeric value of the column. The lengths (number of digits) of the search string and of the numeric column are not significant. Similarly, the data type of the numeric column is not significant. For example, a search string of 123 matches a DECIMAL column containing 00123, or a SMALLINT column containing 123. A search string of 123 does not match a DECIMAL or SMALLINT column containing 12300 (because the numeric values are different). An unsigned value in the string is considered to be positive.

Related references

Finding data within specific columns on page 147

Related references

CHANGE primary command on page 928
EXCLUDE primary command on page 943
FIND primary command on page 953

Changing the view of displayed data

You can change the view of data displayed in a Z Data Tools/Db2 editor session by editing the template to:

- Show only selected columns
- Change the sequence of displayed columns
- Change column headings
- Change the width of columns
- Show leading zeros (numeric columns)

After you have edited the template, you can either save it for future use, or just use it for the current ZDT/Db2 editor session only.

To edit the template in the current ZDT/Db2 editor session, use the TEDIT primary command.

To edit the template from the Table View or Table Edit panel, select the **Edit template** option.

Related tasks

Working with templates on page 74

Related references

TEDIT primary command on page 1006

Selecting a display format

In a Z Data Tools/Db2 editor session, you can view the data in the following display formats:

- Tabular format (TABL), or
- Single-row format (SNGL)

To select the display format you want, you can:

- Enter one of the following values (or just its first letter) in the **Format** field (in the upper right corner of the Table View or Table Edit panel):
  - TABL
  - SNGL
- Use the FORMAT primary command to toggle between TABL and SNGL display formats.
- In TABL display format, use the FS primary command to change to SNGL display format.
- In SNGL display format, use the FT primary command to change to TABL display format.
- If in TABL display format, use the FS prefix command to change to SNGL display format.

The following pages describe the TABL and SNGL display formats.

Related references

FORMAT primary command on page 962
FS primary command on page 965
FT primary command on page 966
Table Edit panel on page 868

TABL display format

In TABL display format (the default format), the rows of the table are presented formatted into columns displayed horizontally across the panel.

Figure 49: View panel in TABL display format (with all optional lines displayed) on page 154 shows an example of a view display in TABL display format.

Figure 49. View panel in TABL display format (with all optional lines displayed)

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                  Table View                    42 rows fetched
TABLE HFMUSER.EMP                                                Format TABL
EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE   JOB
#1     #2           #3      #4              #5       #6      #7         #8   +
CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE       CH(8)
PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+----> <---+-
****  Top of data  ****
000010 CHRISTINE<   I       HAAS<           A00      3978    01.01.1965 PRES
000020 MICHAEL<     L       THOMPSON<       B01      3476    10.10.1973 MANAGE
000030 SALLY<       A       KWAN<           C01      4738    05.04.1975 MANAGE
000050 JOHN<        B       GEYER<          E01      6789    17.08.1949 MANAGE
000060 IRVING<      F       STERN<          D11      6423    14.09.1973 MANAGE
000070 EVA<         D       PULASKI<        D21      7831    30.09.1980 MANAGE
000090 EILEEN<      W       HENDERSON<      E11      5498    15.08.1970 MANAGE
000100 THEODORE<    Q       SPENSER<        E21      0972    19.06.1980 MANAGE
000110 VINCENZO<    G       LUCCHESI<       A00      3490    16.05.1958 SALESR
000120 sean<                O'CONNELL<      A00      2167    05.12.1963 CLERK
000130 DOLORES<     M       QUINTANA<       C01      4578    28.07.1971 ANALYS
000140 HEATHER<     A       NICHOLLS<       C01      1793    15.12.1976 ANALYS
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 50: Edit panel in TABL display format (with all optional lines displayed) on page 154 shows an example of an edit display in TABL display format.

Figure 50. Edit panel in TABL display format (with all optional lines displayed)

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                   Table Edit                     42 rows fetched
TABLE HFMUSER.EMP                                              Format TABL
EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
000000 ****  Top of data  ****
000000 ****  Top of data  ****
000001 000010 CHRISTINE<   I       HAAS<           A00      3978    01.01.1965
000002 000020 MICHAEL<     L       THOMPSON<       B01      3476    10.10.1973
000003 000030 SALLY<       A       KWAN<           C01      4738    05.04.1975
000004 000050 JOHN<        B       GEYER<          E01      6789    17.08.1949
000005 000060 IRVING<      F       STERN<          D11      6423    14.09.1973
000006 000070 EVA<         D       PULASKI<        D21      7831    30.09.1980
000007 000090 EILEEN<      W       HENDERSON<      E11      5498    15.08.1970
000008 000100 THEODORE<    Q       SPENSER<        E21      0972    19.06.1980
000009 000110 VINCENZO<    G       LUCCHESI<       A00      3490    16.05.1958
000010 000120 sean<                O'CONNELL<      A00      2167    05.12.1963
000011 000130 DOLORES<     M       QUINTANA<       C01      4578    28.07.1971
000012 000140 HEATHER<     A       NICHOLLS<       C01      1793    15.12.1976
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Multiple rows are displayed on the one panel (limited by the screen size). Each column of the table is formatted according to the data type for that column. For example, numeric data types are displayed as whole or decimal numbers, and date and time fields are displayed as dates or times.

By default, the headings are the names of the columns as defined in the Db2® catalog. You can change them by editing the template.

For SELECT statement View or Edit only, the heading might be displayed as COL*nnn* where *nnn* is 001, 002, and so on. This indicates a generated column name. Both of the following examples cause ZDT/Db2 to generate a column name:

```
SELECT 1, dbname FROM SYSIBM.SYSTABLES
```

```
SELECT CONCAT(dbname, tsname) FROM SYSIBM.SYSTABLES
```

One or more columns in the ZDT/Db2 editor session might be protected. This can occur if:

- The object being edited has a primary key, and the **Allow updates to primary key** option is not selected.
- The object being edited is a read-only view.
- The object being edited is the result table from an arbitrary SELECT statement (that is, select statement edit).
- The object being edited is a table, but contains columns, excluding ROWID columns, marked as not updateable in the Db2® catalog. The Db2® catalog tables fall into this category.

If you display data in TABL display format, you can scroll up, down, left, and right.

To display the data in hexadecimal format, use the HEX primary command:

```
HEX ON
```

In TABL display format, the view or edit display always shows the column names for each displayed column (or the alternative field heading if you have defined one).

Optionally, you can display up to 3 additional information lines between the column heading line and the top of data. These are (in order):

- A line containing the column number (as defined in the Db2® catalog).
- A line containing the data type for the column (and the length for non-numeric fields).
- A line showing a ruler above each column. The ruler can include offset information and, optionally, information about primary and foreign keys, and unique indexes.

You can select which, if any, of these additional information lines are displayed on the editor display either:

- For *all* ZDT/Db2 editor sessions by setting the appropriate global option using the Editor Options panel, or
- For the *current* ZDT/Db2 editor session only:
  - For View, you set the appropriate option using the View (option 1) panel.
  - For Edit, you set the appropriate option using the Edit (option 2) panel.

Related tasks

Working with templates on page 74
Scrolling on page 160
View and Edit options (options 1 and 2) on page 67
ROWID columns on page 107
Setting options for the current ZDT/Db2 editor session on page 126
Excluding rows on page 162

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616
Editor Options (1 of 8) panel on page 644
HEX primary command on page 966

SNGL display format

In SNGL display format, only one row of the table is displayed on the panel at the one time.

Figure 51: Edit panel in SNGL display format (with all optional columns displayed) on page 156 shows an example of an edit display in SNGL display format.

Figure 51. Edit panel in SNGL display format (with all optional columns displayed)

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Table Edit                          1 of 42
TABLE HFMUSER.EMP                                                  Format SNGL
Top Column is 1    of 13    in Row 1
Ref Rdf Key   Column     Type(len)   Data
#1       PU   EMPNO      CH(6)       000010
#2            FIRSTNME   VC(12)      CHRISTINE<
#3            MIDINIT    CH(1)       I
#4            LASTNAME   VC(15)      HAAS<
#6            PHONENO    CH(4)       3978
#7            HIREDATE   DATE        01/01/1965
#8            JOB        CH(8)       PRES
#9            EDLEVEL    SMINT           18
#10           SEX        CH(1)       F
#11           BIRTHDATE  DATE        08/14/1933
#12           SALARY     DEC(9,2)       52750.00
#13           BONUS      DEC(9,2)        1000.00
#14           COMM       DEC(9,2)        4220.00
****  End of record  ****

Command ===>                                                  Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Previous F11=Next     F12=Cancel
```

The columns of the table are displayed vertically on the panel, with the column header on the left of the panel, and the data for that column on the right. Each column of the table is formatted according to the data type for that column. For example, numeric data types are displayed as whole or decimal numbers, and date and time fields are displayed as dates or times.

By default, the headings are the names of the columns as defined in the Db2® catalog. You can change them by editing the template.

For SELECT statement in a Z Data Tools/Db2 editor session, the heading might be displayed as COL*nnn* where *nnn* is 001, 002, and so on. This indicates a generated column name. Both of the following examples cause ZDT/Db2 to generate a column name:

```
SELECT 1, dbname FROM SYSIBM.SYSTABLES
```

```
SELECT CONCAT(dbname, tsname) FROM SYSIBM.SYSTABLES
```

One or more columns in the ZDT/Db2 editor session might be protected. This can occur if:

- The object being edited has a primary key, and the *"Allow updates to primary key"* option is not selected.
- The object being edited is a read-only view.

- The object being edited is the result table from an arbitrary SELECT statement (that is, select statement edit).
- The object being edited is a table, but contains columns, excluding ROWID columns, marked as not updateable in the Db2® catalog. The Db2® catalog tables fall into this category.

If you display data in SNGL display format, you can:

- Scroll within the row (up and down)
- Scroll forward and backward within the table (to show the next and previous non-excluded rows)

To display the data in hexadecimal format, use the HEX primary command `HEX ON`.

In SNGL display format, the view or edit display always shows two columns: the name of the column, and the data for that column.

Optionally, you can display up to three additional information columns. Two of these columns, **Ref** and **Key**, are displayed to the left of the column containing the column name; the third, **Type(Len)** is displayed between the name and data columns. If you choose to show all information columns, the displayed columns are (in order of appearance from left to right):

Ref

An optional column showing the column number (as defined in the Db2® catalog).

You can turn this optional column on and off with the REFS primary command. The value shown can be zero if displaying the result table from an arbitrary select statement (that is, select statement edit). This indicates either that the column is generated, or that the column is not a column of the first table named in the FROM clause of the SELECT statement that produced the result table.

Rdf

Unused.

Key

An optional column showing information about primary keys, foreign keys, unique indexes and check constraints.

You can turn this optional column on and off with the INDEXINF primary command.

Column

A column showing the column name.

This column may also show the start and end position of the character data displayed for a row. For this to occur:

1. The length of the data for the column must be long enough to require two or more rows to display.
2. The *"Show char position range"* editor option needs to be selected, or the CHARPOS command can be issued to turn on the character position range display. For more information, see Editor Options (3 of 8) panel on page 655 and CHARPOS command CHARPOS primary command on page 935.

Type(Len)

An optional column showing the data type for the column (and the length for non-numeric fields).

You can turn this optional column on and off with the TYPE primary command

Data

A column showing the (formatted) data for the column.

You can customize the view or edit display either:

- For *all* ZDT/Db2 editor sessions by setting the appropriate global option using the Editor Options panels, or
- For the *current* ZDT/Db2 editor session only:
  - For View, you set the appropriate option using the Db2® View panel.
  - For Edit, you set the appropriate option using the Db2® Edit panel.

Related tasks

Working with templates on page 74
Scrolling on page 160

Related references

Db2 View panel on page 631
Db2 Edit panel on page 616
Editor Options (1 of 8) panel on page 644
HEX primary command on page 966
INDEXINF primary command on page 970
REFS primary command on page 994
TYPE primary command on page 1008

Zooming in to see all of a row

You can *"zoom in"* on a single row by pressing the Zoom function key (F2) or by using the ZOOM primary command.

In TABL display format, to display all data in a particular row without having to scroll left or right, move the cursor to the row, then press the Zoom function key (F2). ZDT/Db2 limits the display to just that row. In SNGL display format, just press the Zoom function key (F2).

If you zoom in on a row, only that row is displayed in *"zoomed"* SNGL display format. The zoomed display automatically scrolls to the field that was at the left of the panel, or, if applicable, the column where the cursor was positioned before zooming.

In *"zoomed"* SNGL display format, the word **Zoom** is displayed next to **Format** in the upper right corner of the panel as shown in Figure 52: Zooming in on one row ("zoomed" SNGL display format) on page 159.

Figure 52. Zooming in on one row (“zoomed” SNGL display format)

```
  Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Table Edit                          1 of 42
TABLE HFMUSER.EMP                                            Zoom Format SNGL
                                    Top Column is 1    of 14    in Row 1
Ref Key    Column     Type(len)        Data
#1  PU     EMPNO      CH(6)            000050
#2         FIRSTNME   VC(12)           JOHN<
#3         MIDINIT    CH(1)            B
#4         LASTNAME   VC(15)           GEYER<
#5    NF   WORKDEPT   CH(3)            E01
#6         PHONENO    CH(4)            6789
#7         HIREDATE   DATE             08/17/1949
#8         JOB        CH(8)            MANAGER
#9         EDLEVEL    SMINT                16
#10        SEX        CH(1)            M
#11        BIRTHDATE  DATE             09/15/1925
#12        SALARY     DEC(9,2)            40175.00
#13        BONUS      DEC(9,2)              800.00
#14        COMM       DEC(9,2)             3214.00
****  End of record  ****
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

If you view a record in *“zoomed”* SNGL display format, ZDT/Db2 displays all columns, regardless of whether they were selected for display in the template. ZDT/Db2 displays the columns in the order defined in the template. Any template re-sequencing is ignored. In this display format, you cannot navigate between rows or use the TEDIT primary command.

In TABL or normal *“un-zoomed”* SNGL display format, Z Data Tools only shows the columns selected for display in the template.

For instance, in the example shown in Figure 52: Zooming in on one row (“zoomed” SNGL display format) on page 159, the columns WORKDEPT and BIRTHDATE are shown in *“zoomed”* SNGL display format, even though these columns happen to be *“deselected”* in the template. In *“un-zoomed”* SNGL display format, the same row is displayed:

```
  Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Table Edit                          1 of 42
TABLE HFMUSER.EMP                                              Format SNGL
                                    Top Line is 1    of 12    in Row 1
Ref Rdf Key    Column     Type(len)   Data
#2             FIRSTNME   VC(12)      JOHN<
#3             MIDINIT    CH(1)       B
#4             LASTNAME   VC(15)      GEYER<
#6             PHONENO    CH(4)       6789
#7             HIREDATE   DATE        08/17/1949
#8             JOB        CH(8)       MANAGE
#9             EDLEVEL    SMINT          16
#10            SEX        CH(1)       M
#12            SALARY     DEC(9,2)       40175.00
#13            BONUS      DEC(9,2)         800.00
#14            COMM       DEC(9,2)        3214.00
****  End of record  ****


Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Previous F11=Next     F12=Cancel
```

In *“un-zoomed”* SNGL display format, you can navigate between rows using the PREVIOUS and NEXT primary commands or the Previous function key (F10) and the Next function key (F11).

To return to the display format that was current before zooming in, enter the ZOOM primary command again, or the CANCEL, END, or EXIT command.

Related tasks

TABL display format on page 153
SNGL display format on page 156

Related references

ZOOM primary command on page 1016
TEDIT primary command on page 1006

Scrolling

To scroll through your data, press the following function keys:

Press this function key

To…

Up function key (F7)

Scroll backward (up)

Down function key (F8)

Scroll forward (down)

In TABL display format:

Right function key (F11)

Scroll right

Left function key (F10)

Scroll left

In SNGL display format:

Next function key (F11)

Go to the next row

Previous function key (F10)

Go to the previous row

You can also scroll using the primary commands UP, DOWN, LEFT, RIGHT, TOP, BOTTOM, NEXT and PREVIOUS.

Related tasks

TABL display format on page 153
SNGL display format on page 156
Holding columns on the display when scrolling on page 100

Related references

ZOOM primary command on page 1016
TEDIT primary command on page 1006
UP primary command on page 1009
DOWN primary command on page 939
LEFT primary command on page 971
RIGHT primary command on page 996
TOP primary command on page 1007
BOTTOM primary command on page 924
NEXT primary command on page 983
PREVIOUS primary command on page 988

Controlling how far you scroll

To control how far you scroll when you press one of the scrolling function keys, enter a scroll amount in the **Scroll** field:

Scroll amount

Scrolls…

PAGE

One panel at a time

HALF

Half a panel at a time

DATA

One line or panel column less than a screen at a time

CSR

To the cursor position (if the cursor is not positioned on a row when the scroll is performed, the default scroll amount is PAGE)

nnnn

A number of panel columns (if scrolling left or right) or rows (when scrolling up or down) at a time

 **Note:**



1. In TABL display format, you can temporarily override the amount in the **Scroll** field by typing a scroll amount on the command line, then pressing a scroll function key. For example, if you enter `8` on the command line, then press the Right function key (F11), ZDT/Db2 scrolls right 8 panel columns.
2. In SNGL display format, you can specify the number of rows ZDT/Db2 scrolls forward or backward by typing a number on the command line, then pressing the Next function key (F11) or the Previous function key (F10). For example, if you enter `5` on the command line, then press the Next function key (F11), ZDT/Db2 scrolls forward 5 rows.
3. In TABL display format, the position that the cursor occupies after you scroll left or right depends on the data type of the Db2® column that occupies the resultant panel column:
    - If the data type of the Db2® column is non-numeric, scrolling left or right moves to the relative panel column within the Db2® column (for small columns, scrolling moves to the start of the column).
    - If the data type of the Db2® column is numeric, scrolling left or right moves to the start of the Db2® column.

Scrolling to the first or last row or column

To scroll to the first or last row or column in the table, type `MAX` (or `M`) on the command line, then press one of the scroll function keys. For example, typing `M` then pressing the Right function key (F11) scrolls right to the last column.

You can also scroll to the first or last row by entering `TOP` or `BOTTOM` on the command line.

Excluding rows

When you are viewing or editing data, you can *"exclude"* certain rows in the data you are viewing. These rows are referred to as *excluded rows* and are not displayed on your screen. The remaining rows (those that *are* displayed on your screen) are referred to as *non-excluded rows*.

You may choose to exclude certain rows to restrict the scope of the CHANGE, DELETE, and FIND primary commands. When you use one of these commands with data that contains excluded rows and non-excluded rows, you can specify whether the command is to affect:

- Only non-excluded rows,
- Only excluded rows, or
- All rows

To exclude certain rows in the table you are viewing, you can use:

- The EXCLUDE primary command. For example, the following command excludes all rows with the string *"SMITH"* in column number 3:

  ```
  EXCLUDE ALL SMITH #3
  ```

  The following command excludes the next row containing the string *"XXX"* within either column number 20 or column number 30:

```
EXCLUDE XXX #20 #30
```

You can exclude a range of rows identified by a *"from"* label and a *"to"* label. For example, the following command excludes a range of rows, starting with the row labeled *".LABA"* and ending with the row labeled *".LABB"*:

```
EXCLUDE ALL .LABA .LABB
```

In a Z Data Tools/Db2 editor session, you can specify an asterisk (*) as the search string to find the same string as you specified on the previous EXCLUDE primary command.

In a Z Data Tools/Db2 editor session, you can limit the effect of the EXCLUDE command to specific portions of the data you are searching:

- To limit the search for a string to only where it appears as a *prefix* in the data, specify the PREFIX parameter.
- To limit the search for a string to only where it appears as a *suffix* in the data, specify the SUFFIX parameter.
- To limit the search for a string to only where it appears as a *word* in the data, specify the WORD parameter.

- The following prefix commands:

  X

  Exclude one row.

  X*n*

  Exclude *n* rows.

  XX

  Exclude a block of rows.

You can use the following line commands to show various rows from a block of excluded rows:

F

Show the first row from a block of excluded rows.

F*n*

Show the first *n* rows from a block of excluded rows.

L

Show the last row from a block of excluded rows.

L*n*

Show the last *n* rows from a block of excluded rows.

Related tasks

Showing or hiding excluded rows on page 164

Related references

EXCLUDE primary command on page 943
CHANGE primary command on page 928

DELETE primary command on page 936
FIND primary command on page 953

Showing or hiding excluded rows

In TABL display format, you can choose to hide excluded rows from display, or represent them by shadow lines:

```
- - - - - - - - - - - - - - - - - - - - - - n line(s) excluded
```

The SHADOW primary command controls whether shadow lines are shown or hidden. To turn on shadow lines for excluded rows, enter `SHADOW ON`. To turn off shadow lines for excluded rows, enter `SHADOW OFF`.

In SNGL display format, shadow lines are not displayed. If you use the Next function key (F11) or Previous function key (F10) to scroll through rows in SNGL display format, the only indication that an excluded row has been skipped is that the row number (shown in the upper right corner of the panel) skips accordingly.

To redisplay excluded rows, enter the following command:

```
RESET EXCLUDED
```

Related tasks

TABL display format on page 153
SNGL display format on page 156

Related references

SHADOW primary command on page 1000
RESET primary command on page 994

Sorting the data

To sort the data you are viewing or editing in TABL display format, use the SORT primary command.

You can specify up to 5 column numbers (the most significant first) and whether you want the data sorted in ascending (the default sequence) or descending sequence for each column number you specify.

Examples

`SORT #7 D #6 #3`

Sorts all rows on the column with column number #7 (primary sort key) in descending order, then on the column with column number #6 (secondary sort key) in ascending order, then on the column with column number #3 (minor sort key) in ascending order,

`SORT #6 X`

Sorts all excluded rows on the column with column number #6 in ascending order (as if non-excluded rows did not exist).

Note: The SORT command only changes the order in which you view the data; it has no effect on the order in which the data is stored within Db2®.

Related references

SORT primary command on page 1003

Displaying data in hexadecimal format

To display the data you are browsing or editing in hexadecimal format, use the HEX primary command.

For example, if your display looks something like this:

```
       NAME               CREATOR  TYPE   DBNAME   TSNAME     DBID   OBID
       #1                 #2       #3     #4       #5           #6     #7
       VARCHAR(18)        CH(8)    CH(1)  CH(8)    CH(8)     SMINT  SMINT
       <---+----1----+--> <---+--> -      <---+--> <---+--> <---+> <---+>
000000 ****  Top of data  ****
000001 SYSCOPY            SYSIBM   T      DSNDB06  SYSCOPY   00006  00046
000002 SYSFIELDS          SYSIBM   T      DSNDB06  SYSDBASE  00006  00021
```

then issuing the command HEX ON causes the hexadecimal representation to be displayed:

```
       NAME               CREATOR  TYPE   DBNAME   TSNAME     DBID   OBID
       #1                 #2       #3     #4       #5           #6     #7
       VARCHAR(18)        CH(8)    CH(1)  CH(8)    CH(8)     SMINT  SMINT
       <---+----1----+--> <---+--> -      <---+--> <---+--> <---+> <---+>
000000 ****  Top of data  ****
000001 SYSCOPY            SYSIBM   T      DSNDB06  SYSCOPY   00006  00046
       EEECDDE00000000000 EEECCD44 E      CEDCCFF4 EEECDDE4     00     02
       282367800000000000 28292400 3      42542060 28236780     06     0E

000002 SYSFIELDS          SYSIBM   T      DSNDB06  SYSDBASE  00006  00021
       EEECCCDCE000000000 EEECCD44 E      CEDCCFF4 EEECCCEC     00     01
       282695342000000000 28292400 3      42542060 28242125     06     05
```

Related references

HEX primary command on page 966

Managing the prefix area

The prefix area displays line numbers that represent the row number in the Db2® table or view being edited.

The prefix area also doubles as the prefix command entry area. You enter prefix commands by overtyping the row number for the appropriate row.

You can use prefix commands to:

- Insert or delete rows
- Repeat (duplicate) rows
- Move rows

- Shift data
- Limit the data being edited

In a Z Data Tools/Db2 editor session, to see which prefix commands are valid for the current environment, enter an "?" in the prefix area (against any listed line). ZDT/Db2 displays a list of the line commands you can enter, like those shown in Figure 53: Using the ? prefix command to display a list of valid prefix commands on page 166.

Figure 53. Using the ? prefix command to display a list of valid prefix commands

```
 Process   Options   Utilities   Help
                   ZDT/Db2 Available Edit Prefix Commands
                                                                More:     +
Prefix commands

You can enter the following commands in the prefix area of the data display:

A     identifies the record after which data is to be moved or copied.

B     identifies the record before which data is to be moved or copied.

C     copy one record.
Cn    copy n records.
CC    copy block of records. Mark start and end of block.

D     delete one record.
Dn    delete n records.
DD    delete block of records. Mark start and end of block.

E     displays information about why Db2 rejected the change to a row

F     display the first record of a block of excluded records.
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

In a Z Data Tools/Db2 editor session, you can choose whether a prefix area is displayed next to the data or not.

To specify the default position and display state of the prefix area for *all* ZDT/Db2 editor sessions, set these editor options on the relevant Editor Options panel (0.2):

- **Display prefix area**
- **Display prefix area on the right**
- **Prefix area width (A,6,7,8,9)**

To specify the default position and display state of the prefix area for the *current* ZDT/Db2 editor session, use the PREFIX primary command. For example:

- To display the prefix area on the right, enter `PREFIX RIGHT`.
- To display the prefix area as an 8-digit field, enter `PREFIX 8`.
- To display the prefix area with a width that automatically adjusts (from 6 to 9 digits wide) in order to display the record number, enter `PREFIX A`.
- To remove the display of the prefix area, enter `PREFIX OFF`.

Related references

PREFIX primary command on page 987

Displaying the column number

In SNGL display format, use the REFS primary command to show or hide the column number for each displayed column:

- If the column numbers are not currently displayed, ZDT/Db2 displays an additional column (to the left of the column containing the column name and, if displayed, to the left of the column showing index information) showing the column number for each column.

  If you enter the REFS primary command in TABL display format, the display is unaltered. However, if you change to SNGL display format, the display status of the additional column is reversed.
- If the column numbers are already displayed, the REFS primary command removes them from display.

Note: You can also display the column numbers by selecting the **Show column number** option (for TABL, SNGL, or both display formats).

Related references

REFS primary command on page 994
Editor Options (3 of 8) panel on page 655

Displaying the data type

In SNGL display format, use the TYPE primary command to show or hide the data type (and, for non-numeric columns, the length of the column) for each displayed column:

- If the data type details are not currently displayed, ZDT/Db2 displays an additional column (between the name and data columns) showing the data type details for each column.

  If you enter the TYPE primary command in TABL display format, the display is unaltered. However, if you change to SNGL display format, the display status of the additional column is reversed.
- If the data type details are already displayed, the TYPE primary command removes them from display.

Note: You can also display the data type details by selecting the **Show data type** option (for TABL, SNGL, or both display formats).

Related references

REFS primary command on page 994
Editor Options (3 of 8) panel on page 655

Handling special data

The following sections describe how to work with special types of data.

DATE and TIME columns

Db2® accepts dates and times in a number of formats.

DATE columns

In a Z Data Tools/Db2 editor session, the display format for DATE columns depends on the setting of the DATE parameter in the Db2® DSNHDECP installation module. For further information, see your Db2® systems administrator, the *Db2 UDB for z/OS SQL Reference*, or other Db2® manuals.

If you edit a DATE column, you can enter a date in any format recognized by Db2®. ZDT/Db2 converts the date you enter into the appropriate display format. For example, to set the contents of a DATE column to represent September 27, 2004, you can enter `27.09.2004`.

However, you need to be careful when choosing the separator character. If you try to enter `27/09/2004`, ZDT/Db2 interprets *"27"* as a month and rejects the date as incorrect. A more subtle example is 08/05/1900 and 08.05.1900. ZDT/Db2 interprets the first as 5 August, 1900, but the second as May 8, 1900.

TIME columns

In a Z Data Tools/Db2 editor session, the display format for TIME columns depends on the setting of the **External Format for TIME Data Type Columns** System Option.

If you edit a TIME column, you can enter a time in any format recognized by Db2®. ZDT/Db2 converts the time you enter into the appropriate display format.

If you enter times in USA format (`hh.mm AM` or `PM`), the absence of *second* information means that the time is stored internally as hh.mm.00 (that is, the seconds portion is always set to zero). This can cause unexpected results when comparing different TIME columns that have the same display format but different internal representations with respect to seconds.

Related tasks

ZDT/Db2 system options (option 0.2) on page 65

Related references

Systems Options (1 of 4) panel on page 857

Large numeric fields

If ZDT/Db2 generates a template, it sets the column width for numeric columns such that any value can be correctly displayed.

However, if as a result of editing the template, a numeric column contains a valid numeric value, but the value is too large to fit in the width allocated to the column (as defined in the template), the value is truncated. The column is highlighted to indicate that truncation has occurred.

To display the entire column without truncation, you need to edit the template, and specify an output width for the column:

1. Enter `TEDIT` on the command line to display the Column Selection/Edit panel.
2. Enter `E` template editor prefix command in the Prefix Command area to display the Column Attributes panel.
3. In the **Output width** field, enter a value that is wide enough to display the entire column.
4. Press the Exit function key (F3) key until you are back at the editor panel.

Related references

Column Selection/Edit panel on page 497
EXCLUDE primary command on page 943
FIND primary command on page 953
TEDIT primary command on page 1006

Working with large tables

When operating in view mode, ZDT/Db2 allows you to process large Db2® tables without the need to specify a large TSO region size. This feature is not available if the product installer has chosen to disable it.

A number of options and processes associated with working with large tables are available and described in these sections:

- SQL optimization options on page 169
- Large table options on page 170
- Data sampling on page 170

SQL optimization options

You set the **Optimization options** on the on the Editor Options (2 of 8) panel to specify a `OPTIMIZE FOR nnn ROWS` clause in the SQL statement used to retrieve data from Db2®.

Table 6: Effect of SQL optimization options on page 169 describes the behavior for various combinations of options:

Table 6. Effect of SQL optimization options

Use row count value	Row count	Optimize for … Rows	Behavior
Not selected	0	0	No clause
Not selected	0	*nnn*	OPTIMIZE FOR *nnn* ROWS appended
Not selected	*nnn*	0	No clause
Not selected	*nnn*	*mmm*	OPTIMIZE FOR *mmm* ROWS appended
Selected	0	0	No clause
Selected	0	*nnn*	OPTIMIZE FOR *nnn* ROWS appended
Selected	*nnn*	0	OPTIMIZE FOR *nnn* ROWS appended

Table 6. Effect of SQL optimization options (continued)

Use row count value	Row count	Optimize for … Rows	Behavior
Selected	*nnn*	mmm	OPTIMIZE FOR *nnn* ROWS appended

Related references

Editor Options (2 of 8) panel on page 649

Large table options

You can set the **Scrollable cursor type** option on the Editor Options (2 of 8) panel to either 1 (Insensitive) or 2 (Sensitive static). When processing a large Db2® object in place holder mode only, ZDT/Db2 uses the appropriate type of scrollable cursor. There is no simple method for you to determine which type of scrollable cursor is in use. A DBG trace shows the SQL statement used to declare the cursor and fetch the data.

Related references

Editor Options (2 of 8) panel on page 649

Data sampling

Data sampling allows you specify that a subset of the rows in a Db2® object are loaded into the editor. This can be particularly useful when you are working with very large Db2® tables and require a representative sample of the data.

The **Sampling type** option on the Editor Options (2 of 8) panel allows you to specify how you want ZDT/Db2 to select rows from a Db2® object in order to provide sample data:

Clustered sampling

A specified number of rows are loaded into the editor.

Random sampling

Each row of the object is considered for inclusion in the editor. A random number generator and a frequency are used to determine whether the row is actually loaded.

Each of these two methods of selecting sample data are discussed here in detail.

Clustered data sampling

To specify clustered data sampling, set the **Sampling type** option on the Editor Options (2 of 8) panel to **2. Clustered sampling**.

These points describe the behavior of the ZDT/Db2 editor when using clustered data sampling:

- Data sampling applies to browse, view, and edit.
- When using data sampling, the editor always loads all sampled rows into memory. Therefore large table support is NOT available when sampling data.

- The options that are pertinent to clustered sampling are:
  - **Row count** on the function entry panel.
  - **Start position** on the function entry panel.
  - **Sampling limit** on the Editor Options (2 of 8) panel.
  - **Initial skip count** on the Editor Options (2 of 8) panel.
  - **Include count** on the Editor Options (2 of 8) panel.
  - **Skip count** (final) on the Editor Options (2 of 8) panel.

Table 7: Behavior of ZDT/Db2 editor for options related to clustered data sampling on page 171 describes the behavior of the ZDT/Db2 editor for various options related to clustered data sampling.

Table 7. Behavior of ZDT/Db2 editor for options related to clustered data sampling

Row count	Start position	Sampling limit	Initial skip count	Include count	Skip count	Behavior
0	1	0	0	*nnn*	0	All rows sampled
0	1	*bbb*	0	*nnn*	0	*bbb* rows sampled[12].
rrr	1	0	0	*nnn*	0	rrr rows sampled.[12]
rrr	1	*bbb*	0	*nnn*	0	The minimum of (rrr,*bbb*) rows are sampled.[12]
0	*sss*	0	0	*nnn*	0	Rows sampled from *sss* until the end of table.
0	*sss*	*bbb*	0	*nnn*	0	*bbb* rows sampled, commencing *sss*.[12]
rrr	*sss*	0	0	*nnn*	0	rrr rows sampled, commencing *sss*.[12]
rrr	*sss*	*bbb*	0	*nnn*	0	The minimum of (rrr,*bbb*) rows are sampled, commencing *sss*.[12]
0	1	0	*xxx*	*nnn*	*yyy*	Rows are sampled in groups of *xxx*+*nnn*+*yyy* rows. Within each group, *xxx* rows are skipped; *nnn* rows are sampled and *yyy* rows are skipped. The first group starts are row 1, the second group at (*xxx*+*nnn*+*yyy*+1). Groups are sampled until the end of the result table.[12]
0	1	*bbb*	*xxx*	*nnn*	*yyy*	Rows are sampled in groups of *xxx*+*nnn*+*yyy* rows. Within each group, *xxx* rows are skipped; *nnn* rows are sampled and *yyy* rows are skipped. The first group starts are row 1, the

12. Sampling continues until one of these conditions is met:
    - Any non-zero **Sampling limit** is reached.
    - Any non-zero **Row count** limit is reached.
    - The end of the result table is reached.

Table 7. Behavior of ZDT/Db2 editor for options related to clustered data sampling

(continued)

Row count	Start position	Sampling limit	Initial skip count	Include count	Skip count	Behavior
						second group at (*xxx*+*nnn*+*yyy*+1). Groups are sampled until *bbb* rows are loaded..[12]
rrr	1	0	*xxx*	*nnn*	*yyy*	Rows are sampled in groups of *xxx*+*nnn*+*yyy* rows. Within each group, *xxx* rows are skipped; *nnn* rows are sampled and *yyy* rows are skipped. The first group starts are row 1, the second group at (*xxx*+*nnn*+*yyy*+1). Groups are sampled until rrr rows are loaded.[12]
rrr	1	*bbb*	*xxx*	*nnn*	*yyy*	Rows are sampled in groups of *xxx*+*nnn*+*yyy* rows. Within each group, *xxx* rows are skipped; *nnn* rows are sampled and *yyy* rows are skipped. The first group starts are row 1, the second group at (*xxx*+*nnn*+*yyy*+1). Groups are sampled until the minimum of (*bbb*,rrr) rows are loaded.[12]
rrr	*sss*	*bbb*	*xxx*	*nnn*	*yyy*	Rows are sampled in groups of *xxx*+*nnn*+*yyy* rows. Within each group, *xxx* rows are skipped; *nnn* rows are sampled and *yyy* rows are skipped. The first group starts are row *sss*, the second group at *sss*+(*xxx*+*nnn*+*yyy*). Groups are sampled until the minimum of (*bbb*,rrr) rows are loaded.[12]

Random data sampling

To specify random data sampling, set the **Sampling type** option on the Editor Options (2 of 8) panel to **3. Random sampling**.

These points describe the behavior of the ZDT/Db2 editor when using random data sampling:

- Data sampling applies to browse, view, and edit.
- When using data sampling, the editor always loads all sampled rows into memory. Therefore large table support is NOT available when sampling data.
- The options that are pertinent to random sampling are:
  - **Row count** on the function entry panel.
  - **Start position** on the function entry panel.
  - **Sampling limit** on the Editor Options (2 of 8) panel.
  - **Sampling frequency** on the Editor Options (2 of 8) panel.
  - **Sampling seed** on the Editor Options (2 of 8) panel.

Table 8: Behavior of ZDT/Db2 editor for options related to random data sampling on page 173 describes the behavior of the ZDT/Db2 editor for various options related to random data sampling.

Table 8. Behavior of ZDT/Db2 editor for options related to random data sampling

Row count	Start position	Sampling limit	Sampling frequency	Sample seed	Behavior[13]
0	1	0	0.*fff*	0	Rows are fetched, starting at row 1 and continuing until the end of the result table.
0	1	*bbb*	0.*fff*	0	Rows are fetched, starting at row 1 and continuing until the end of the result table, or until *bbb* rows have been added to the editor.
rrr	1	0	0.*fff*	0	*rrr* rows are fetched, starting at the first row. Some subset (approximately 0.*fff* x *rrr*) of these rows is added to the sample data set.

13. 1. Sampling continues until one of the following conditions is met:
    - Any non-zero Sampling limit is reached.
    - Any non-zero Row count (fetch) limit is reached.
    - The end of the result table is reached.

    The Sampling limit sets an upper bound on the number of rows loaded into the editor. This is the number of rows in an editor session. To reach this many rows, approximately 1/(Sampling frequency x Sampling limit) rows must be fetched. By contrast, the Row count limit sets an upper bound on the number of rows that are fetched from the object. The number of rows that are sampled is approximately Row count x Sampling frequency.

    With "small" sampling frequencies, specifying a low Row count limit may result in no rows being sampled. For example, with a Row count limit of 60, a Sampling count of 2000, and a Sampling frequency of 0.01, there is a reasonable chance that no rows are sampled.

    2. A Sampling frequency of 0.*fff* results in, on average and for large numbers of rows sampled, 0.*fff* x 100% of rows being sampled. Therefore a frequency of 0.1 results in 10% of rows being sampled. For small frequencies, many rows need to be processed to find each matching row. For any particular random sample of data there is no guarantee that the number of rows in the sample will exactly reflect the sampling frequency.

    When a Sampling seed value of 0 is specified, ZDT/Db2 uses the fraction part of the second's value derived from the current system clock value to initiate the random number generator. This value is accurate to the microsecond; therefore each seed has a value in the range 0–999999 inclusive. It is unlikely that two random samples generated with a sampling seed of 0 will be identical.

    When the same user-specified Sampling seed is used for the same table and other conditions, the data samples produced will be identical.

Table 8. Behavior of ZDT/Db2 editor for options related to random data sampling

(continued)

Row count	Start position	Sampling limit	Sampling frequency	Sample seed	Behavior[13]
rrr	1	*bbb*	0.*fff*	0	At most *rrr* rows are fetched, starting at the first row. Some subset (approximately 0.*fff* x *rrr*) of these rows, but at most *bbb* rows, is added to the sample data set.
0	*sss*	0	0.*fff*	0	Rows are sampled, starting at row *sss*, and continuing until the end of the result table.
0	*sss*	*bbb*	0.*fff*	0	Rows are fetched, starting at row *sss*, and continuing until the end of the result table or until *bbb* rows have been added to the editor.
rrr	*sss*	0	0.*fff*	0	*rrr* rows are fetched, starting at row *sss*. Some subset (approximately 0.*fff* x *rrr*) of these rows is added to the sample data set.

13. 1. Sampling continues until one of the following conditions is met:
    - Any non-zero Sampling limit is reached.
    - Any non-zero Row count (fetch) limit is reached.
    - The end of the result table is reached.

    The Sampling limit sets an upper bound on the number of rows loaded into the editor. This is the number of rows in an editor session. To reach this many rows, approximately 1/(Sampling frequency x Sampling limit) rows must be fetched. By contrast, the Row count limit sets an upper bound on the number of rows that are fetched from the object. The number of rows that are sampled is approximately Row count x Sampling frequency.

    With "small" sampling frequencies, specifying a low Row count limit may result in no rows being sampled. For example, with a Row count limit of 60, a Sampling count of 2000, and a Sampling frequency of 0.01, there is a reasonable chance that no rows are sampled.

    2. A Sampling frequency of 0.*fff* results in, on average and for large numbers of rows sampled, 0.*fff* x 100% of rows being sampled. Therefore a frequency of 0.1 results in 10% of rows being sampled. For small frequencies, many rows need to be processed to find each matching row. For any particular random sample of data there is no guarantee that the number of rows in the sample will exactly reflect the sampling frequency.

    When a Sampling seed value of 0 is specified, ZDT/Db2 uses the fraction part of the second's value derived from the current system clock value to initiate the random number generator. This value is accurate to the microsecond; therefore each seed has a value in the range 0–999999 inclusive. It is unlikely that two random samples generated with a sampling seed of 0 will be identical.

    When the same user-specified Sampling seed is used for the same table and other conditions, the data samples produced will be identical.

Table 8. Behavior of ZDT/Db2 editor for options related to random data sampling

(continued)

Row count	Start position	Sampling limit	Sampling frequency	Sample seed	Behavior[13]
rrr	*sss*	*bbb*	0.*fff*	0	At most *rrr* rows are fetched, starting at row *sss*. Some subset (approximately 0.*fff* x *rrr*) of these rows, but at most *bbb* rows, is added to the sample data set.
0	1	0	0.*fff*	0	Rows are fetched, starting at row 1 and continuing until the end of the result table. The random number generator starts with seed *yyy*.
0	1	*bbb*	0.*fff*	0	Rows are fetched, starting at row 1 and continuing until the end of the result table, or until *bbb* rows have been added to the editor. The random number generator starts with seed *yyy*.
rrr	1	0	0.*fff*	0	*rrr* rows are fetched, starting at the first row. Some subset (approximately 0.*fff* x *rrr*) of these rows is added to the sample data set. The random number generator starts with seed *yyy*.

13. 1. Sampling continues until one of the following conditions is met:
    - Any non-zero Sampling limit is reached.
    - Any non-zero Row count (fetch) limit is reached.
    - The end of the result table is reached.

    The Sampling limit sets an upper bound on the number of rows loaded into the editor. This is the number of rows in an editor session. To reach this many rows, approximately 1/(Sampling frequency x Sampling limit) rows must be fetched. By contrast, the Row count limit sets an upper bound on the number of rows that are fetched from the object. The number of rows that are sampled is approximately Row count x Sampling frequency.

    With "small" sampling frequencies, specifying a low Row count limit may result in no rows being sampled. For example, with a Row count limit of 60, a Sampling count of 2000, and a Sampling frequency of 0.01, there is a reasonable chance that no rows are sampled.

    2. A Sampling frequency of 0.*fff* results in, on average and for large numbers of rows sampled, 0.*fff* x 100% of rows being sampled. Therefore a frequency of 0.1 results in 10% of rows being sampled. For small frequencies, many rows need to be processed to find each matching row. For any particular random sample of data there is no guarantee that the number of rows in the sample will exactly reflect the sampling frequency.

    When a Sampling seed value of 0 is specified, ZDT/Db2 uses the fraction part of the second's value derived from the current system clock value to initiate the random number generator. This value is accurate to the microsecond; therefore each seed has a value in the range 0–999999 inclusive. It is unlikely that two random samples generated with a sampling seed of 0 will be identical.

    When the same user-specified Sampling seed is used for the same table and other conditions, the data samples produced will be identical.

Table 8. Behavior of ZDT/Db2 editor for options related to random data sampling

(continued)

Row count	Start position	Sampling limit	Sampling frequency	Sample seed	Behavior[13]
rrr	1	*bbb*	0.*fff*	0	At most *rrr* rows are fetched, starting at the first row. Some subset (approximately 0.*fff* x *rrr*) of these rows, but at most *bbb* rows, is added to the sample data set. The random number generator starts with seed *yyy*.
0	*sss*	0	0.*fff*	0	Rows are sampled, starting at row *sss* and continuing until the end of the result table. The random number generator starts with seed *yyy*.
0	*sss*	*bbb*	0.*fff*	0	Rows are fetched, starting at row *sss* and continuing until the end of the result table, or until *bbb* rows have been added to the editor. The random number generator starts with seed *yyy*.

13. 1. Sampling continues until one of the following conditions is met:
    - Any non-zero Sampling limit is reached.
    - Any non-zero Row count (fetch) limit is reached.
    - The end of the result table is reached.

    The Sampling limit sets an upper bound on the number of rows loaded into the editor. This is the number of rows in an editor session. To reach this many rows, approximately 1/(Sampling frequency x Sampling limit) rows must be fetched. By contrast, the Row count limit sets an upper bound on the number of rows that are fetched from the object. The number of rows that are sampled is approximately Row count x Sampling frequency.

    With "small" sampling frequencies, specifying a low Row count limit may result in no rows being sampled. For example, with a Row count limit of 60, a Sampling count of 2000, and a Sampling frequency of 0.01, there is a reasonable chance that no rows are sampled.

    2. A Sampling frequency of 0.*fff* results in, on average and for large numbers of rows sampled, 0.*fff* x 100% of rows being sampled. Therefore a frequency of 0.1 results in 10% of rows being sampled. For small frequencies, many rows need to be processed to find each matching row. For any particular random sample of data there is no guarantee that the number of rows in the sample will exactly reflect the sampling frequency.

    When a Sampling seed value of 0 is specified, ZDT/Db2 uses the fraction part of the second's value derived from the current system clock value to initiate the random number generator. This value is accurate to the microsecond; therefore each seed has a value in the range 0–999999 inclusive. It is unlikely that two random samples generated with a sampling seed of 0 will be identical.

    When the same user-specified Sampling seed is used for the same table and other conditions, the data samples produced will be identical.

Table 8. Behavior of ZDT/Db2 editor for options related to random data sampling

(continued)

Row count	Start position	Sampling limit	Sampling frequency	Sample seed	Behavior[13]
rrr	*sss*	0	0.*fff*	0	*rrr* rows are fetched, starting at row *sss*. Some subset (approximately 0.*fff* x *rrr*) of these rows is added to the sample data set. The random number generator starts with seed *yyy*.
rrr	*sss*	*bbb*	0.*fff*	0	At most *rrr* rows are fetched, starting at row *sss*. Some subset (approximately 0.*fff* x *rrr*) of these rows, but at most *bbb* rows, is added to the sample data set. The random number generator starts with seed *yyy*.

Related references

Db2 Browse panel on page 612
Db2 Edit panel on page 616

13. 1. Sampling continues until one of the following conditions is met:
    - Any non-zero Sampling limit is reached.
    - Any non-zero Row count (fetch) limit is reached.
    - The end of the result table is reached.

    The Sampling limit sets an upper bound on the number of rows loaded into the editor. This is the number of rows in an editor session. To reach this many rows, approximately 1/(Sampling frequency x Sampling limit) rows must be fetched. By contrast, the Row count limit sets an upper bound on the number of rows that are fetched from the object. The number of rows that are sampled is approximately Row count x Sampling frequency.

    With "small" sampling frequencies, specifying a low Row count limit may result in no rows being sampled. For example, with a Row count limit of 60, a Sampling count of 2000, and a Sampling frequency of 0.01, there is a reasonable chance that no rows are sampled.

    2. A Sampling frequency of 0.*fff* results in, on average and for large numbers of rows sampled, 0.*fff* x 100% of rows being sampled. Therefore a frequency of 0.1 results in 10% of rows being sampled. For small frequencies, many rows need to be processed to find each matching row. For any particular random sample of data there is no guarantee that the number of rows in the sample will exactly reflect the sampling frequency.

    When a Sampling seed value of 0 is specified, ZDT/Db2 uses the fraction part of the second's value derived from the current system clock value to initiate the random number generator. This value is accurate to the microsecond; therefore each seed has a value in the range 0–999999 inclusive. It is unlikely that two random samples generated with a sampling seed of 0 will be identical.

    When the same user-specified Sampling seed is used for the same table and other conditions, the data samples produced will be identical.

Db2 View panel on page 631
Editor Options (2 of 8) panel on page 649

Null values and VARCHAR processing

ZDT/Db2 provides different options for processing null values, and varying-length columns.

Null column indicator character

ZDT/Db2 provides options that you can use to specify two null column characters:

Null column indicators: Input

For use when editing a null column

Null column indicators: Display

Indicates a null column when it is displayed

In a Z Data Tools/Db2 editor session, entering the null column input indicator character in the first (leftmost) column of a column that permits null values indicates that the column is to have a null value. For ZDT/Db2 to recognize the null column input indicator character, you must type the selected character in the first position of the input field, and remove all the remaining characters.

ZDT/Db2 also provides an option whereby it interprets a column containing spaces as containing a null value.

Related references

Editor Options (4 of 8) panel on page 662

Varying-length columns

You can use ZDT/Db2 options to specify two string delimiter characters:

String delimiters: Input

For use when editing a varying-length column

String delimiters: Display

Indicates the end of a varying-length column when it is displayed

You can also select an option (**Remove trailing spaces**) to control how ZDT/Db2 interprets trailing spaces in a varying-length column.

The following examples show the effect of the **Remove trailing spaces** option, and the use of the string delimiter character ("#" in this case), if you specify a string (for a varying-length column) in a Z Data Tools/Db2 editor session. In the examples, ">" and "<" indicate the start and end of the data you type.

- If the **Remove trailing spaces** option is selected, trailing spaces are removed:

  ```
  Example string #<
  ```

  Stored as “Example string”.

- If the **Remove trailing spaces** option is not selected, trailing spaces are not removed:

  ```
  Example string #<
  ```

  Stored as “Example string ”.

Related tasks

View and Edit options (options 1 and 2) on page 67
Setting options for the current ZDT/Db2 editor session on page 126

Related references

Editor Options (5 of 8) panel on page 667

Unicode data

The ZDT/Db2 editor handles character data stored in Unicode differently to data stored in EBCDIC or ASCII. The differences are summarized here.

ZDT/Db2 normally retrieves character data from Db2® with automatic conversion to the CCSID of the ZDT/Db2 plan, normally CCSID 37. This is US EBCDIC; other EBCDIC CCSIDs may also be used.

When ZDT/Db2 retrieves character data stored in Unicode (within Db2®), this automatic data conversion does not occur. The ZDT/Db2 editor processes the character data internally in native Unicode format. This means that, prior to display on the terminal, the data is converted from Unicode to the CCSID of the terminal. This may result in conversion errors when a Unicode character has no corresponding code point in the CCSID of the terminal. Any characters that cannot be converted are shown on the display as periods, indicating an unconverted character. When data displayed (and possibly changed) on the terminal is processed, it is converted from the CCSID of the terminal into Unicode, prior to submission to Db2®. This conversion will always be successful, since Unicode includes code points for all characters in commonly used CCSIDs.

Data corruption is possible when the displayed data includes a period indicating an unconverted Unicode character, and the data for the column is changed by overtyping. Any periods in the modified data are converted to the Unicode equivalent. Therefore, if the period represents an unconverted Unicode character, that character will be corrupted by the change. When character data is displayed in hexadecimal mode, the hexadecimal characters normally represent the encoding for the displayed character in the CCSID of the terminal. For example, 'A' is represented by 'C1'x in EBCDIC. When the character data is for a Unicode-encoded column, the hexadecimal characters represent the native Unicode data, not the encoding of the displayed character in the CCSID of the terminal. For example, 'A' is represented by '41'x when Unicode data is being displayed.

You can also make changes to the data in a column stored in Unicode by turning hexadecimal display on (HEX ON command) and overtyping the hexadecimal characters directly. When you do this, you should use the Unicode representation of any character, not the terminal's CCSID representation.

Assuming the target column is character data stored within Db2® in Unicode, the FIND, CHANGE, and EXCLUDE commands operate like this:

- Strings such as 'ABC' are converted to Unicode prior to any search or change occurring.
- Hexadecimal strings such as '414243'x are *not* converted to Unicode prior to any search or change occurring. Therefore `F 'ABC'` and `F '414243'x` are equivalent commands when the target column contains Unicode data, but not when the target column contains EBCDIC data.

Changing data

In a Z Data Tools/Db2 editor session, you can change data by:

- Overtyping
- Using the CHANGE primary command
- Entering prefix commands in the prefix area next to each row

Related tasks

Installation settings that affect CAPS on page 180
Changing data by overtyping on page 181
Changing data with the CHANGE command on page 182
Changing data using prefix commands on page 186

Installation settings that affect CAPS

The setting of CAPS determines whether or not the characters you enter in a Z Data Tools/Db2 editor session are converted to uppercase. If CAPS is set to ON, ZDT/Db2 translates all data in rows that are changed by overtyping from lowercase to uppercase, and translates replacement data specified in the CHANGE command from lowercase to uppercase. If CAPS is OFF, the new data you type is not converted to uppercase.

When ZDT/Db2 is installed, a number of customization settings affect CAPS. These settings determine:

- The initial setting for CAPS (ON or OFF)
- Whether you can change this setting by means of the edit options; that is, whether the setting is *"updateable"* (meaning you can change it) or *"fixed"* (meaning you cannot change it)
- Whether you are permitted to issue the ZDT/Db2 CAPS primary command

For more information about these customization settings, see the *Customization Guide*.

For the first and second of the above settings, you can verify their value by looking at the third Editor Options panel.

For the third of the above settings, if you try to issue the CAPS command but ZDT/Db2 has been customized so that the CAPS command is unavailable, ZDT/Db2 displays the message *"The CAPS command has been disabled at installation time."*.

If the installation option for CAPS is *"fixed"*, each ZDT/Db2 editor session starts with the installation setting for CAPS (either ON or OFF) and you cannot change the setting using the edit options.

If the installation option for CAPS is *"updateable"*, you can change the setting (to either ON or OFF) using the edit options and each ZDT/Db2 editor session starts with the current CAPS setting.

If the CAPS command is disabled, and the installation option for CAPS is *"fixed"*:

- If the installation option for CAPS is ON, to enter lowercase data you must first issue the CASE MIXED command
- If the installation option for CAPS is OFF, to enter uppercase data you must use the Shift or Caps Lock keys

Related tasks

Changing data by overtyping on page 181
Changing data with the CHANGE command on page 182
Converting data to uppercase on page 189

Related references

Editor Options (1 of 8) panel on page 644
CAPS primary command on page 926
CASE primary command on page 927

Changing data by overtyping

You can change the data displayed in a Z Data Tools/Db2 editor session by overtyping it with the new data.

To successfully change data by overtyping:

- The new data you enter must be compatible with the data type of the column being changed. For example, you can only enter numeric values in a column that has a numeric data type.
- The column must be updateable (as defined in the Db2® catalog).
- If the table is subject to referential integrity, the new data must not cause a referential integrity violation.

Related tasks

"Read-only" views, non-updateable columns on page 190
Editing related tables on page 192
Converting data to uppercase on page 189

Related references

Db2 data types on page 440

Changing data with the CHANGE command

When you use the CHANGE command to change data, you must specify one or more column numbers to be included in the search (in the same way as you use the FIND command to find data).

For example, the following command changes all occurrences of the string *"CHAIR"* in all rows to the string *"TABLE"* in either of the columns with column numbers #5 and #6.

```
CHANGE ALL CHAIR TABLE #5,#6
```

And the following command changes the next occurrence of the value 400 in column number 3 to 500:

```
CHANGE 400 500 #3
```

To change the next occurrence of the string, press the RChange function key (F6). To find the next occurrence of the string and optionally change it, use a combination of the RFind function key (F5) and the RChange function key (F6).

To find the same string as you specified on the previous CHANGE primary command, specify an asterisk (*) as the search string.

To use the same replacement string as you specified on the previous CHANGE primary command, specify an asterisk (*) as the replacement string.

Further examples

`CHANGE BLACK WHITE #4 #7`

In TABL display format, changes the next occurrence of *"BLACK"* to "WHITE", where *"BLACK"* is completely contained within either of the columns represented by column numbers 4 and 7.

In SNGL display format, changes the next occurrence of *"BLACK"* to *"WHITE"*, where *"BLACK"* is completely contained within either of the columns represented by column numbers 4 and 7, in the current row only.

`CHANGE BLACK WHITE ALL #2 #8`

In TABL display format, changes all occurrences of *"BLACK"* to *"WHITE"*, where *"BLACK"* is completely contained within either of the columns represented by column numbers 2 and 8, in all rows.

In SNGL display format, changes all occurrences of *"BLACK"* to *"WHITE"*, where *"BLACK"* is completely contained within either of the columns represented by column numbers 2 and 8, in the current row only.

Related concepts

Installation settings that affect CAPS on page 180

Related tasks

Changing data within excluded or non-excluded rows on page 185
Changing data within a label range on page 186
Finding data within specific columns on page 147

Related references

CHANGE primary command on page 928
Table Edit panel on page 868

"From" and "To" strings of different lengths

For character columns (CHAR, VARCHAR), you can specify *"from"* and *"to"* strings (*from_string* and *to_string*) of different lengths, subject to the following conditions:

- For fixed-length character columns, when the *"to"* string (*to_string*) is shorter than the *"from"* string (*from_string*), ZDT/Db2 performs the change and pads the column with spaces. When the *"to"* string (*to_string*) is longer than the *"from"* string (*from_string*), ZDT/Db2 performs the change only if there are enough unused characters in the column to accommodate the change.

  For example, for a given row, if column #2 is a fixed-length, character column of length 8 containing *"GEORGE "* (with two trailing spaces):

  `CHANGE GEORGE FRED #2`

  Changes *"GEORGE "* (with two trailing spaces) to *"FRED "* (with four trailing spaces) in that row.

  `CHANGE GEORGE WILLIAM #2`

  Changes *"GEORGE "* (with two trailing spaces) to *"WILLIAM "* (with one trailing space) in that row.

  `CHANGE GEORGE ALEXANDER #2`

  ZDT/Db2 does not perform the change to that row as there are not enough unused characters in the column to accommodate the change.
- For varying-length character columns, when the *"to"* string (*to_string*) is shorter than the *"from"* string (*from_string*), ZDT/Db2 performs the change and adjusts the length of the column downwards by the difference between the lengths of *to_string* and *from_string*. When the *"to"* string (*to_string*) is longer than the *"from"* string (*from_string*), ZDT/Db2 performs the change only if there are enough unused characters in the column to accommodate the change. In this case, ZDT/Db2 adjusts the length of the column upwards by the difference between the lengths of *from_string* and *to_string*.

Handling long strings

For character columns (CHAR, VARCHAR), you can specify *"from"* and *"to"* strings (*from_string* and *to_string*) up to 100 characters long. However, when you are dealing with long strings, you may find that you cannot fit the whole CHANGE command (including other parameters) on the command line. To overcome the problem, do one of the following to display an Extended Command Entry pop-up panel (see Figure 54: Extended Command Entry pop-up panel for the CHANGE command on page 184):

- Enter `cx` (for Change Extended)
- Enter the CHANGE primary command (or one of its abbreviations, such as C) *with no parameters*

Note: If you have previously used the CHANGE or FIND commands in the current ZDT/Db2 editor session, to display the Extended Command Entry pop-up panel in this way, you will need to first use the RESET command before entering the CHANGE command (or one of its abbreviations) without any parameters. Alternatively, enter `CX`.

The Extended Command Entry pop-up panel contains five lines (each 50 characters long) that you use to enter all the CHANGE command parameters (but not the actual keyword CHANGE or any of its abbreviations). ZDT/Db2 treats the five lines on the pop-up panel as contiguous, allowing you to specify long *"from"* and *"to"* strings, the Db2® column number and optionally any other parameters, using up to 250 characters in total.

Note: You can continue a string (or any other parameter) from the last position on one line of the pop-up panel to the first position of the next line. However, this does not mean that you need to completely fill a line before using the next line. For example, you can specify the *"from"* string on the first of the five lines, the *"to"* string on the next line, and the Db2® column number (specified as *#n*) on the next line. If you do not use all of a line, ZDT/Db2 treats the unused part of the line as a single space.

Figure 54: Extended Command Entry pop-up panel for the CHANGE command on page 184 shows the Extended Command Entry pop-up panel for the CHANGE command. In this example, any previous occurrences of the string *"dichlorodiphenyltrichloroethane"* in Db2® columns 4 or 7, between the labels .abc and .def, are to be changed to the string *"trinitrophenylmethylnitramine"* for non-excluded rows only.

Figure 54. Extended Command Entry pop-up panel for the CHANGE command

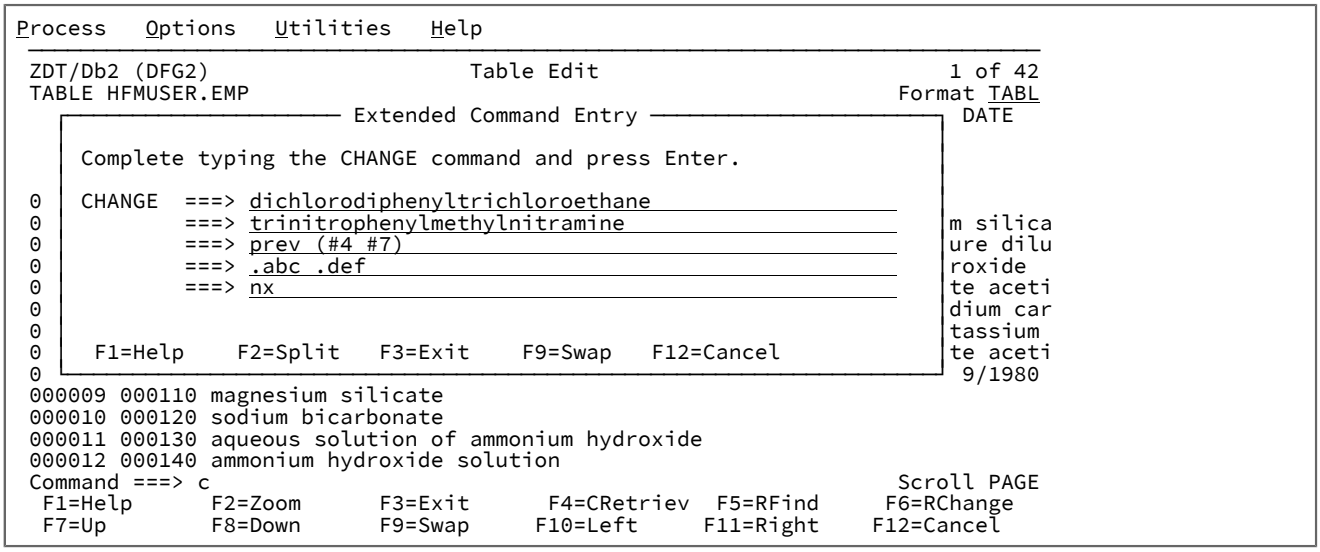
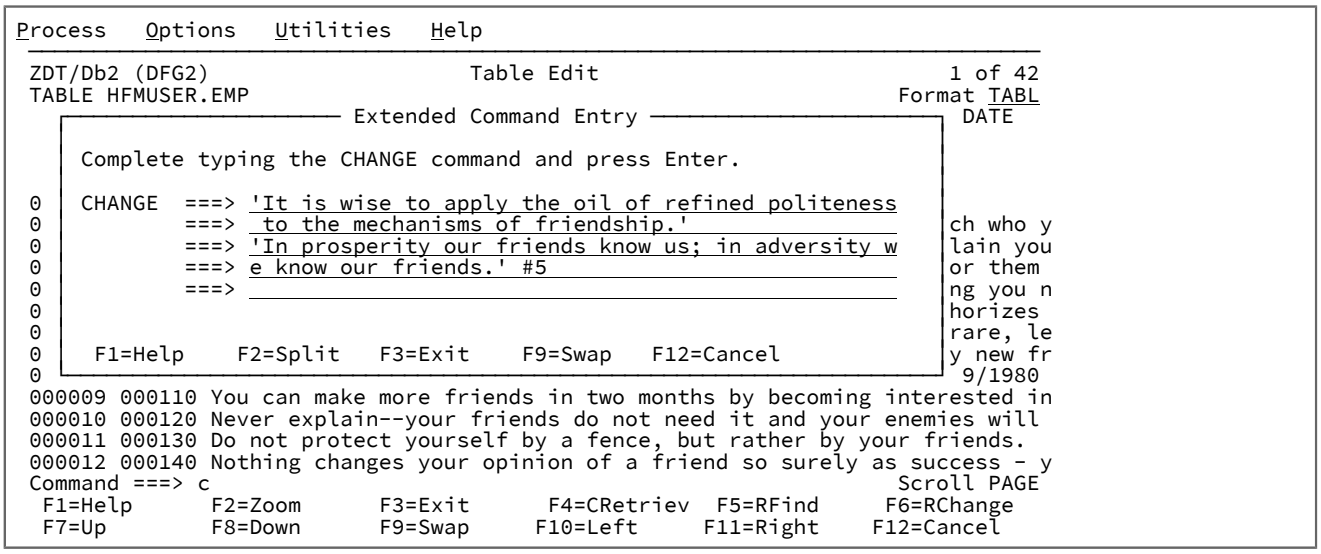


Figure 55: Extended Command Entry pop-up panel for the CHANGE command showing contiguous nature of entry lines on page 185 shows the pop-up panel for the CHANGE command where the next occurrence of a long string (84 characters) in Db2® column 5 is to be changed to another long string (70 characters).

Figure 55. Extended Command Entry pop-up panel for the CHANGE command showing contiguous nature of entry lines



Changing numeric columns

If you use the CHANGE command to change a numeric column, then the lengths of *from_string* and *to_string* are not significant. For example, if column #2 is a numeric column, then the following CHANGE command is permitted, because *from_string* and *to_string* are interpreted as numeric values:

```
CHANGE 3.1415 3.2 #2
```

Changing data within excluded or non-excluded rows

If your data contains excluded and non-excluded rows, you can limit the effect of the CHANGE command to:

- Only non-excluded rows (parameter NX), or
- Only excluded rows (parameter EX or X)

For example, the following command changes all occurrences of the string *"GREEN"*, in column number 2 or column number 5 in non-excluded rows only, to the string *"BLACK"*:

```
CHANGE ALL GREEN BLACK (#2 #5) NX
```

And the following command changes the next occurrence of the value 250, in column number 2 in an excluded row, to 520:

```
CHANGE 250 520 #2 EX
```

Related tasks

Excluding rows on page 162

Related references

CHANGE primary command on page 928

Changing data within a label range

You can limit the effect of the CHANGE command to a range of rows, identified by a pair of labels indicating the first and last rows to be searched for the data you want to change.

For example, in the range of rows from the label .LABA to the end of the data, the following command changes the next occurrence of the string *"EVIL"* in column number 4 to the string *"GOOD"*:

```
CHANGE EVIL GOOD  #4 .LABA .ZLST
```

And, in the range of rows from the current cursor position to the label .LABB, the following command changes all occurrences of the value 48 in column number 3 or column number 11 to the value 148:

```
CHANGE ALL 48 148 #3,#11 .ZCSR .LABB
```

Related tasks

Assigning labels to rows on page 150

Related references

CHANGE primary command on page 928

Limiting the change to specific portions of the data

You can limit the effect of the CHANGE command to specific portions of the data you are searching:

- To limit the search for a string to only where it appears as a *prefix* in the data, specify the PREFIX parameter.
- To limit the search for a string to only where it appears as a *suffix* in the data, specify the SUFFIX parameter.
- To limit the search for a string to only where it appears as a *word* in the data, specify the WORD parameter.

Related references

CHANGE primary command on page 928

Changing data using prefix commands

In a Z Data Tools/Db2 View or Edit editor session, you can enter prefix commands in the prefix area of one or more rows on the display. The prefix area is typically represented by a column of 6-digit numbers on the far left or far right of your display.

You can use prefix commands to:

- Insert or delete rows
- Repeat (duplicate) rows
- Move rows
- Overlay data in one or more rows with data from one or more other rows
- Shift data

- Copy or move rows to and from a clipboard
- Limit the data being edited

To set the position of the prefix area (left or right), or turn the prefix area on or off, use the PREFIX primary command.

You can use most of the prefix area commands on either a single row or a block of consecutive rows:

- To perform an operation on a single row, you enter the appropriate command code.
- To perform an operation on a block of consecutive rows, either enter the command code preceded or followed by the number of lines, or enter the appropriate block command code at the start and end lines of the block.

Generally, you need to type over only the first 1 or 2 characters of the line number to enter a prefix command. Sometimes, however, typing a single character can be ambiguous. In the following example, it is unclear whether the intended prefix command is R to repeat line 31700, or R3 to repeat the line three times:

```
031600
R31700
031800
```

In such cases, the editor assumes that you have not typed a number following the prefix command (it cannot detect if you overtype a character with the same character). If you want to repeat the line three times, you can use any of the following procedures:

- Type one or more blanks following the R3:

  ```
  R3 700
  ```

- Type `R3` and press the Erase EOF key to clear the rest of the Line Command field, or press the Erase EOF key and then type `R3`.
- Type one or more blanks after the R but before the number such that the number when entered is different than the characters being overtyped.:

  ```
  R 3700
  ```

- Type the number before the R, ensuring that the number when entered is different than the characters being overtyped:

  ```
  3R1700
  ```

To clear any outstanding commands in the prefix area, use the RESET primary command.

Note: Moving rows in a Z Data Tools/Db2 editor session does not affect the order in which the rows are stored in the underlying Db2® table.

Overlaying data in existing rows

You can overlay data in a row with data from another row by specifying the O or OO prefix commands.

When data is to be copied or moved by the C (copy) or M (move) prefix commands and overlaid on one or more existing rows, the O or OO (overlay) prefix commands specify the destination for the data. You can only overlay data onto character columns (CHAR and VARCHAR).

The data that is copied or moved overlays blanks in the destination rows.

If the destination column is VARCHAR:

- It must have a non-zero length.
- If the length of the data being moved or copied is longer than the destination column, only the data that can fit into the destination column is moved or copied.

When data is to be moved or copied and then overlaid on a destination:

- Where the destination is a *single row*:
  - The O (overlay) prefix command specifies the destination for the data.

    You can type a number after the O prefix command to specify the number of times that the M or C prefix command is to be performed. For example, typing the command `O3` against a row causes the data to be moved or copied and then overlaid on that row and also the next two rows.
- Where the destination is a *block of rows*:
  - The OO (overlay, multiple-line target) prefix command specifies the first and last row of the destination for the data.

To overlay one or more *single* rows:

1. Type either `M` or `C` in the prefix area of the row that is to be moved or copied.
2. Type `O` in the prefix area of the row that the moved or copied record is to overlay.
3. Press Enter. The data being moved or copied overlays the specified row or rows.

To overlay one or more *blocks* of rows:

1. Type either `MM` or `CC` in the prefix area of the first and last rows of a block of rows that is to be moved or copied. You can scroll (or use FIND or LOCATE) between typing the first command and the second command, if necessary.
2. Type `OO` in the prefix area of the first and last rows that the block of rows being moved or copied is to overlay. Again, you can scroll (or use FIND or LOCATE) between typing the first `OO` and the second `OO`, if necessary.
3. Press Enter. The rows that contain the two CC or MM commands and all of the rows between them overlay the rows that contain the two OO commands and all of the rows between them.

The number of source and receiving rows need not be the same. If there are more receiving rows, the source rows are repeated until the receiving rows are gone. If there are more source rows than receiving rows, the extra source rows are ignored.

Related topics

Table Edit panel on page 868

Related references

Table Edit panel on page 868
PREFIX primary command on page 987
RESET primary command on page 994

Converting data to uppercase

In a Z Data Tools/Db2 editor session, the CAPS and CASE primary commands control whether data in the data area is converted to uppercase on input (CAPS, CAPS ON or CASE UPPER), or left alone (CASE, CASE MIXED or CAPS OFF; this is the default action).

You can abbreviate the CAPS command to CAP and the CASE command to CAS.

The CAPS ON and CASE UPPER commands have slightly different effects.

CAPS or CAPS ON is similar to the ISPF editor CAPS command:

- If you overtype any data in a row, only the current column is converted to uppercase.
- The CHANGE command treats the new (replacement) string as if it were specified in all uppercase unless:
  - The new string is enclosed in single quotation marks, prefixed by the letter C (for example, `C'New String'`), or
  - The new string is specified as a hexadecimal string (for example, `X'C1C2C3'`)

  If either of the previous conditions apply, the new string is not converted to uppercase.

CAPS or CAPS ON affects only rows changed after the CAPS or CAPS ON command is entered.

CASE UPPER converts all data in any new or changed rows to uppercase, regardless of how the rows are changed. CASE UPPER affects all rows inserted or changed in the current ZDT/Db2 editor session, including rows inserted or changed before the CASE UPPER command is entered.

The CAPS OFF, CASE, and CASE MIXED commands have the same effect: they turn off conversion to uppercase.

Any CAPS command overrides any previously entered CASE command, and any CASE command overrides any previously entered CAPS command.

Note:

1. In the ISPF editor with CAPS ON, a line is considered changed if you type *anything* on the line (if you overtype a space with a space, it is considered to be a change). However, in the ZDT/Db2 editor, overtyping a character with the same character is not considered a change.
2. The values for CASE and CAPS are not maintained between ZDT/Db2 editor sessions. Each new ZDT/Db2 editor session begins with CAPS OFF; that is, data entered in lowercase is not converted to uppercase.

Related concepts

Installation settings that affect CAPS on page 180

Related tasks

Changing data by overtyping on page 181

Related references

CAPS primary command on page 926
CASE primary command on page 927
Table Edit panel on page 868
Editor Options (1 of 8) panel on page 644

"Read-only" views, non-updateable columns

If the object you are editing is a view, it is possible that one or more columns within the view are marked as non-updateable in the Db2® catalog. In Db2® parlance, such a view is referred to as a *read-only* view.

If you attempt to edit a read-only view that contains *only* non-updateable columns, ZDT/Db2 displays a Browse session (instead of an Edit session) and issues an advisory message at the start of the session. Otherwise (that is, if the read-only view contains at least one updateable column), ZDT/Db2 displays a Z Data Tools/Db2 editor session showing the non-updateable columns as protected and issues an advisory message as the start of the ZDT/Db2 editor session. If you edit a view containing one or more non-updateable columns, not all edit operations are possible.

If you edit a table that contains one or more columns, excluding ROWID columns, that are marked as non-updateable in the Db2® catalog (such as a Db2® catalog table), ZDT/Db2 displays a Z Data Tools/Db2 editor session showing the non-updateable columns as protected and issues an advisory message at the start of the ZDT/Db2 editor session. If you edit a table containing one or more non-updateable columns, excluding ROWID columns, not all edit operations are possible.

Related tasks

ROWID columns on page 107

Materialized query tables

If you want to edit objects of type, *"MATERIALIZED QUERY"*, then the object must be defined with the **DEFERRED** refresh mode and with the **MAINTAINED BY USER** option. If these criteria are not satisfied, the user cannot edit the object, and the VIEW/EDIT session is substituted with a BROWSE session. A message is issued informing the user that the VIEW/EDIT session type is not supported for the object with the current refresh mode and maintenance settings.

Related tasks

ROWID columns on page 107

Deleting rows

In a Z Data Tools/Db2 editor session, you can delete rows by using:

- The DELETE primary command
- The D, D*n*, or DD prefix commands in the prefix area next to each row (TABL display only)

In TABL display format, you can restrict the rows you delete by:

- Limiting the rows that are deleted to only excluded or only non-excluded rows
- Limiting the rows that are deleted to those within a range of rows identified by a pair of labels
- Using the D, D*n*, or DD prefix commands

In SNGL display format, DELETE deletes the currently displayed row only. To delete the currently displayed row, enter:

```
DELETE
```

Note:

1. In TABL display format, care should be taken when using the DELETE primary command to ensure that you only delete the rows you intend to. The command `DELETE ALL` (with no other parameters) deletes all rows in the table or view being edited.
2. If the table target has referential constraints, you might not be able to delete rows. For further information, see Handling errors during an Edit session on page 202.

Related tasks

Deleting rows within excluded or non-excluded rows on page 192
Deleting rows within a label range on page 192

Related references

DELETE primary command on page 936
Table Edit panel on page 868

Deleting rows within excluded or non-excluded rows

If your data contains excluded and non-excluded rows, you can limit the effect of the DELETE command to:

- Only non-excluded rows (parameter NX), or
- Only excluded rows (parameter EX or X)

For example, the following command deletes all excluded rows:

```
DELETE ALL EX
```

And the following command deletes the first non-excluded row:

```
DELETE NX
```

Related tasks

Excluding rows on page 162

Related references

DELETE primary command on page 936

Deleting rows within a label range

You can limit the effect of the DELETE command to a range of rows, identified by a pair of labels indicating the first and last rows to be deleted.

For example, the following command deletes all rows from the label .LABA to the label .LABB.

```
DELETE ALL .LABA .LABB
```

Related tasks

Assigning labels to rows on page 150

Related references

DELETE primary command on page 936

Editing related tables

ZDT/Db2 provides support for editing tables that are subject to referential integrity (RI) constraints. If you are not familiar with the terms *parent table*, *dependent table*, *primary key*, *foreign key* and *referential integrity*, see the relevant Db2® manuals.

Note: The support provided by ZDT/Db2 for editing related tables applies only to Db2® tables, not to any Db2® view created on a table that is subject to RI constraints.

This section describes the behavior of the REDIT command when a Db2® error has been encountered, and the Db2® error is an RI violation. For a description of issuing the REDIT command when no RI error has occurred, see Listing related tables on page 193.

If ZDT/Db2 encounters an error when attempting to save the data changed in an Edit session and the error is due to a RI violation, it automatically determines the name of the other table (the related table) in the RI relationship. You can then start another ZDT/Db2 Edit session for the related table. The new Edit session (dependent session) is displayed as a new ISPF logical session, as if you had typed `START` on the command line and then selected ZDT/Db2 Edit. You can use the ISPF commands SWAP and SWAP LIST to move between different logical sessions.

ZDT/Db2 customizes the template for the new Edit session to make the identification of the key columns easier. If the related table is a dependent table, ZDT/Db2 displays only the rows with values that match the primary key in the parent table. If the related table is a parent table, you can use the IP (Insert primary key) prefix command to insert a row with a primary key matching the foreign key that caused the error.

If an Edit session has active dependent Edit sessions, you cannot save changes made in the session until you end all dependent Edit sessions. If you cancel an edit session that has active dependent Edit sessions, all dependent Edit sessions are also canceled. Any uncommitted changes in any of the dependent Edit sessions are lost.

Listing related tables

When you are in a Z Data Tools/Db2 View or Edit editor session, you can issue the REDIT primary command to display a list of tables related by a Db2® referential integrity (RI) constraint to the Db2® object currently being edited.

When you enter the REDIT primary command, you can optionally supply a field number that identifies the column in the table of the Db2® object currently being edited.

The way in which ZDT/Db2 responds after you enter the REDIT primary command depends on the object you are editing, the position of the cursor, and whether you specify a column number:

- When the object being edited does *not* have RI constraints, a new ZDT/Db2 Edit session is stacked on top of the existing Edit session. You are presented with the Db2® Edit entry panel.
- When the object being edited *does* have RI constraints:
  - When the cursor is not located in the data portion of the screen, or is not located on the data for a column of the table, ZDT/Db2 displays a selection list showing all the Db2® objects that are directly related to the object being edited by an RI constraint. This includes tables that are parent tables in a relationship with the table being edited, and tables that are dependent tables in a relationship with the table being edited.
  - When the cursor is located on the data for a column of the table, and that column is not part of a primary, parent or foreign key, ZDT/Db2 displays a selection list showing all the Db2® objects that are directly related to the object being edited by an RI constraint. This includes tables that are parent tables in a relationship with the table being edited, and tables that are dependent tables in a relationship with the table being edited.

- When the cursor is located on the data for a column of the table, and that column is part of a primary, parent, or foreign key, ZDT/Db2 displays a selection list showing only those Db2® objects that are directly related to the object being edited, limited to those tables where the primary, parent, or foreign key of the table being edited contains the column where the cursor is located.
- When you have specified a field number with the REDIT primary command and the column identified by the field number is part of a primary, parent, or foreign key, ZDT/Db2 displays a selection list showing only those Db2® objects that are directly related to the object being edited, limited to those tables where the primary, parent, or foreign key of the table being edited contains the column identified by the field number.
- When a field number is provided with the REDIT command and the column identified by the field number is not part of a primary, parent, or foreign key, ZDT/Db2 displays a selection list showing all the Db2® objects that are directly related to the object being edited by an RI constraint. This includes tables that are parent tables in a relationship with the table being edited, and tables that are dependent tables in a relationship with the table being edited.

The list of related tables for an object is displayed by the ZDT/Db2 Object List utility on the Related tables panel. You can issue a number of prefix commands against any of the objects shown on the list. For example:

Prefix command

ZDT/Db2 displays…

EE or VE

A new ZDT/Db2 editor session for the selected object. The new ZDT/Db2 editor session appears in a new ISPF logical screen (as though you had split the screen and started a new ZDT/Db2 session).

The first panel displayed, and the rows displayed in the Edit (or View) session, depend on where you position the cursor at the time of entering the REDIT command:

- When the cursor is located on the data for a column that is part of a parent, primary, or foreign key, the new ZDT/Db2 editor session starts with the display of data, and contains only those rows related to the row where the cursor was located.
- When the cursor is not located on the data for a column that is part of a parent, primary, or foreign key, the new ZDT/Db2 editor session starts with the display of the Db2® Edit (or Db2® View) entry panel (where you can alter the object name if required), and contains all rows of the object.

E or V

A new ZDT/Db2 Edit session stacked on top of the existing Edit session.

To see all the available commands, type "?" in the selection field (**SEL**) to the left of any entry and press Enter.

You can only issue the REDIT command when editing a table. The REDIT command is inoperative when editing a view.

Related references

Working with lists of Db2 objects on page 303
Related tables panel on page 787
REDIT primary command on page 992

Self-referencing constraints

A self-referencing constraint exists if a Db2® object is subject to a primary or foreign key relationship in which the parent table and the dependent table are the same table. If the DELETE rule for the relationship is CASCADE, the deletion or change of one row can cause a recursive deletion of other rows in the table. If ZDT/Db2 detects a self-referencing constraint defined on a table, it does not attempt to determine which rows are affected by a deletion or change of a primary key value. This also applies to any other tables affected by other relationships defined on the table.

Referential integrity errors

There are two types of referential integrity errors:

- The first type of referential integrity error occurs if you are editing a parent table and you either delete or update a primary key value. If the restrict rule applies to foreign key values in the dependent table, Db2® rejects the update or delete operation. To make the change to the primary key value in the parent table, you must edit the dependent table and either delete or change all rows with a matching foreign key.
- The second type of referential integrity error occurs if you are editing a dependent table and you create a foreign key value for which there is no corresponding entry in the parent table. Db2® rejects the insert or update operation (because a foreign key value cannot exist without a corresponding primary key value). To change the value of the foreign key in the dependent table, you must edit the parent table and create a row with a primary key value that matches the value of the new foreign key value in the dependent table.

Related tasks

Handling errors during an Edit session on page 202

Starting an Edit session of a related table

To start a Z Data Tools/Db2 Edit session of a related table, perform one of these actions:

- In the information panel for an RI error (accessed by typing `E` in the prefix area for an RI error), type `REDIT` on the command line.
- Use the RE prefix command against a row marked in error if the error is an RI error.

Note: When the object being edited is a Db2® view, both the REDIT line command and the RE prefix command are disabled.

Related references

REDIT primary command on page 992
Table Edit panel on page 868

Differences when you edit a related table

A Z Data Tools/Db2 Edit session for a related table is similar to a typical Edit session, with the following exceptions:

- The new session starts with the display of data for the related table.
- You cannot change the edit options. The current global edit options apply.
- There is no opportunity to edit the template before displaying the data.
- The panel title shows *"Table Edit (related)"* (instead of *"Table Edit"*).
- The location information is replaced with either *"PARNT"* or *"DPDNT"* and the name of the other object in the relationship. *"PARNT"* indicates that the other table is the parent table in the relationship. *"DPNDT"* indicates the other table is the dependent table in the relationship.
- For a dependent table, the template includes a WHERE clause that limits the data displayed to only those rows that match the primary key value in the parent table. If the parent table is subject to a self-referencing constraint, all rows might be displayed.
- The template includes information about the order in which columns are displayed. Columns that comprise the primary or foreign key are displayed (in order) at the left (TABL display format), or top (SNGL display format), of the other object in the relationship.
- In TABL display format, columns that form part of the relationship's key are identified with a *"K"* in the sixth character of the scale header line.
- If an Edit session has active dependent Edit sessions (that is, you have used either the REDIT primary command or the RE prefix command to start another ZDT/Db2 edit session of a related table, *and* that Edit session is still active), you cannot save any changes you have made to data until you end all dependent Edit sessions.
- If you cancel an Edit session that has active dependent Edit sessions, ZDT/Db2 also cancels all dependent Edit sessions. Any uncommitted changes in the canceled dependent Edit sessions are lost.

Related tasks

Self-referencing constraints on page 195

Related references

REDIT primary command on page 992
Table Edit panel on page 868

Differences when saving or canceling while editing a related table

If you have initiated one or more Edit sessions for related tables, the following restrictions apply:

- If there are active dependent Edit sessions (that is, you have used either the REDIT primary command or the RE prefix command to start another ZDT/Db2 Edit session for a related table, *and* any of those edit sessions are still active), you cannot save any changes to data until you have ended all dependent Edit sessions.
- If you cancel an Edit session that has active dependent Edit sessions, any dependent Edit sessions are also canceled. Any uncommitted changes in the canceled dependent Edit sessions are lost.

Related references

REDIT primary command on page 992
Table Edit panel on page 868

Navigating between Edit sessions

To switch between two or more ZDT/Db2 Edit sessions, use the ISPF commands SWAP and SWAP LIST. For more information, see the *z/OS ISPF User's Guide Vol 1*.

If the table you are editing is part of a larger RI structure, you might need to start several Edit sessions of related tables to make the required changes to the original table.

Deleting a primary key

If you delete a row in a parent table, it might generate an error if there are rows in the dependent table and the delete restrict rule is specified for the relationship. The delete restrict rule prevents the deletion of the primary key value unless you first delete (or change) all the rows in the dependent table with foreign key values matching the primary key value.

In this situation:

1. Start a Z Data Tools/Db2 Edit session for the dependent table by either using the RE prefix command, or REDIT primary command from the error information panel. Usually, only those rows that would be affected by the deletion of the primary key in the parent table are displayed (see following note).
2. Either delete the displayed rows (see following note), or change the values of the foreign keys to another valid value.
3. Press the Exit function key (F3) to validate and commit your changes.
4. Return to the Edit session for the primary table and proceed with deletion of the primary key value.

Note:

1. In most cases, ZDT/Db2 shows only those rows that would be affected by the deletion of the primary key in the parent table. The exception is the situation where the parent table has a self-referencing constraint that might cause the deletion of other rows in the parent table if a primary or parent key value is deleted or changed. In this situation, ZDT/Db2 shows all rows and you must select the dependent rows.
2. If the parent or dependent tables are also parent or dependent tables in other relationships, changes to the primary or foreign key values can result in other errors.

Related references

REDIT primary command on page 992
Table Edit panel on page 868

Inserting a new foreign key

If you insert a row in a dependent table, it might generate an error if there is no row in the parent table with a corresponding primary key value. Db2® prevents the insertion of the foreign key value until a primary key is created in the parent table.

The parent table can be the same as the table currently being edited (a self-referencing constraint. This is indicated by the absence of an *"R"* in the rightmost position of the prefix area for the row in error. You can use the E prefix command to display an information panel with details of the relationship name, the parent and dependent table names, and the names of the columns in the key in both the parent and dependent tables.

In the situation where the parent and dependent tables are different:

1. Start a Z Data Tools/Db2 Edit session for the parent table by either using the RE prefix command, or REDIT primary command from the error information panel.
2. Use the IP prefix command to insert a new row in the table, with a primary key that matches the foreign key value entered in the Edit session for the dependent table. You can also change an existing row so that its primary key matches the foreign key entered in the Edit session of the dependent table.
3. Press the Exit function key (F3) to validate and commit your changes.
4. Return to the original Edit session and save the new foreign key value.

Note: If the parent or dependent tables are also parent or dependent tables in other relationships, changes to the primary or foreign key values can result in other errors.

Related tasks

Self-referencing constraints on page 195

Related references

REDIT primary command on page 992
Table Edit panel on page 868

Updating a primary key

Db2® interprets updating a primary key as a deletion, followed by an insertion of a new row the same as the deleted row but with the new primary key value.

If you update a row in a parent table, it can generate an error if there are rows in the dependent table with matching foreign key values and the delete restrict rule is specified for the relationship. The delete restrict rule prevents the deletion of the primary key value unless you first delete (or change) all the rows in the dependent table with foreign key values matching the primary key value.

In this situation:

1. Repeat the row containing the original primary key value you want to change.

   Note: After the repeat operation, the new row is displayed immediately below the repeated row and is intensified.

2. Change the primary key in the *new* row to the required value.
3. Use the SAVE primary command to verify that there are no other Db2® errors.
4. Delete the original row.
5. Start a Z Data Tools/Db2 Edit session for the dependent table by either using the RE prefix command, or REDIT primary command from the error information panel. Usually, only those rows that would be affected by the deletion of the primary key in the parent table are displayed (see following note).
6. Either delete the displayed rows (see following note), or change the values of the foreign keys to another valid value.
7. Press the Exit function key (F3) to validate and commit your changes.
8. Return to the Edit session for the primary table and proceed with deletion of the primary key value.

Note:

1. In most cases, ZDT/Db2 shows only those rows that would be affected by the deletion of the primary key in the parent table. The exception is the situation where the parent table has a self-referencing constraint that might cause the deletion of other rows in the parent table if a primary or parent key value is deleted or changed. In this situation, ZDT/Db2 shows all rows and you must select the dependent rows.
2. If the parent or dependent tables are also parent or dependent tables in other relationships, changes to the primary or foreign key values can result in other errors.

Related references

REDIT primary command on page 992
SAVE primary command on page 999
Table Edit panel on page 868

Adding a primary key

You can add a row with a new primary key value to a table using one of these methods:

- You can insert a new row in the table (I prefix command) and then change the values in the new row to create the required primary key value.
- You can repeat an existing row and then change the values in the new row to create the required primary key value.

> **Note:** If you use this method, you *must* change the new row (added immediately below the original row, and intensified), rather than the original row. The reason for this is that changing the original row instead of the new row results in ZDT/Db2 attempting to update the original row with the new primary key value. This change fails if the original row has dependent rows, and SQLCODE -531 results. ZDT/Db2 also reports a duplicate row error (SQLCODE -803) against the new row as it attempts to insert a new row, identical to the original row.

Related tasks

Updating a primary key on page 199
Handling errors during an Edit session on page 202

Related references

Table Edit panel on page 868

Issuing Db2® commands from within a Z Data Tools/Db2 session

When you are working in a Z Data Tools/Db2 session, you can issue a Db2® command and view the output returned by Db2®.

There are two ways in which you can enter a Db2® command from within a Z Data Tools/Db2 session:

- By displaying the Enter and Execute Db2® Commands panel and entering the required Db2® command (prefixed by "Db2® ") on the panel.
- From some ZDT/Db2 panels, entering the required Db2® command on the command line.

Issuing Db2® commands from the Enter and Execute Db2® Commands panel

To issuing Db2® commands from the Enter and Execute Db2® Commands panel:

1. Display the Enter and Execute Db2® Commands panel by performing one of these actions:
   - Select option **6** (Command) from the Primary Option Menu panel and press Enter.
   - From the Process pull-down menu on the Action bar, select option **1** (Db2® Command) and press Enter.
2. On the Enter and Execute Db2® Commands panel, under **Db2® Command:**, type a hyphen (-) followed immediately by the Db2® command you want to issue.

For example:

```
-DISPLAY ARCHIVE
```

Note: If you do not prefix the command with a hyphen, one is added by ZDT/Db2.

3. When you issue either the DISPLAY THREAD or DISPLAY DATABASE command, you can optionally specify the maximum number of lines to be returned in the **Line count** field.
4. Press Enter.

   ZDT/Db2 displays the output from the Db2® command in an ISPF View session as shown in Figure 56: Example of output displayed by Db2 after Db2 command issued on page 202.

Issuing Db2® commands from the command line

You can issuing a Db2® command from the command line on these ZDT/Db2 panels:

- Table Browse panel
- Table Edit panel
- Table View panel
- Object list panels displayed with the Object List utility
- Privilege details panels
- Enter, Execute and Explain SQL Statements panel

To issue a Db2® command from the command line of a Z Data Tools/Db2 panel:

1. Type `DB2`, followed a space, and then the Db2® command you want to issue.

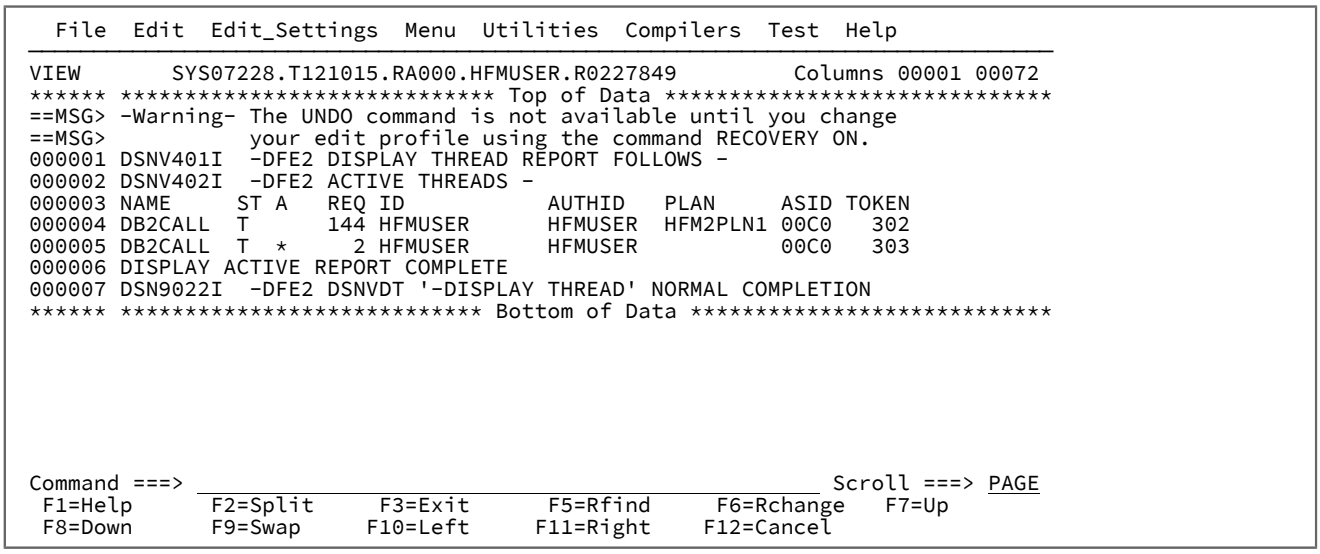
   For example, to issue the Db2® command DISPLAY ARCHIVE, type:

   ```
   DB2 DISPLAY ARCHIVE
   ```

2. Press Enter.

   ZDT/Db2 displays the output from the Db2® command in an ISPF View session as shown in Figure 56: Example of output displayed by Db2 after Db2 command issued on page 202.

Figure 56. Example of output displayed by Db2® after Db2® command issued



Related references

Enter and Execute Db2 Commands panel on page 683

Handling errors during an Edit session

ZDT/Db2 might report errors if it attempts to implement the changes to data that you have made in an Edit session. These errors are detected by Db2® when ZDT/Db2 issues INSERT, UPDATE, or DELETE SQL statements. SQL errors can arise for many reasons, including violation of uniqueness, referential integrity and check data constraints defined on the Db2® object being edited. If you are unfamiliar with these concepts, see the *Db2 UDB for z/OS Administration Guide*.

If ZDT/Db2 detects an error, it marks the row that caused the error with either *"=ERR "*, *"=ERR R"*, or the SQLCODE in the prefix area. The choice of error indicator is determined by the **Show SQLCODE** editor option. Rows that were deleted are reinserted at the top of the current display (in TABL display format). Save operations cannot proceed until all pending errors have been corrected. If the Db2® object you are editing is subject to many constraints, or if many changes have been made to the data without verification, it might be impossible to resolve all errors. In this situation, cancel the Edit session.

Related tasks

View and Edit options (options 1 and 2) on page 67
Setting options for the current ZDT/Db2 editor session on page 126
Ending a Z Data Tools/Db2 editor session on page 133
Example of correcting an error during an Edit session on page 205

Related references

Editor Options (3 of 8) panel on page 655

Determining why an error occurred

You can find more information about the error detected against a row by typing `E` in the prefix field for that row. An error information panel is displayed with an explanation of why Db2® rejected the change operation.

If the SQLCODE is displayed in the prefix area, see the *Db2 UDB for z/OS Messages and Codes* for a detailed explanation of the error.

Some errors are the result of referential integrity (RI) constraint violations. If the RI relationship applies to two tables, these errors are indicated by the presence of an *"R"* in the rightmost position of the prefix area.

You can use the RE prefix command against rows with RI errors that apply to two tables to start another ZDT/Db2 Edit session. The new session is an Edit session for the other table affected by the RI error.

Related tasks

Editing related tables on page 192
Example of correcting an error during an Edit session on page 205

Related references

Table Edit panel on page 868

"Duplicate Row" errors

ZDT/Db2 reports a *"Duplicate Row Error"* if the insert or update operation would have resulted in two rows with the same value with respect to a unique index defined on the table. You can use the Primary Key and Unique Index indicators to identify columns of the table that are part of unique indexes. Note that if you used row selection criteria to limit the amount of data in your Edit session, the duplicate row might not be displayed in your Edit session.

Related tasks

Selecting rows on page 86

"No Primary Key" errors

ZDT/Db2 reports a *"No Primary Key"* error if your change would have created a foreign key value for which there is no corresponding primary key in the parent table of the relationship. This is a *referential integrity* (RI) constraint error. The prefix area error indicator only shows an *"R"* in the rightmost position if the parent table is different to the table currently being edited.

You can use the foreign key indicators to identify columns of the table that are defined as foreign keys.

Note: The foreign key information is only available if the *"Retrieve foreign key information when building templates"* option was selected when the template for the object was built.

The error information panel for this error shows the names of the parent and dependent tables in the relationship that caused the error, and the name of the relationship. At the bottom of the panel are the names of the columns that comprise the key in both the parent and dependent tables, along with the value that caused the error.

If the parent table is different to the dependent table, you can enter the RE prefix command against the row, or enter the REDIT primary command from the error information panel, to edit the parent table of the relationship and correct the error.

Related tasks

Editing related tables on page 192

Related references

Systems Options (1 of 4) panel on page 857
REDIT primary command on page 992

"Delete Restrict Rule" errors

ZDT/Db2 reports a *"Delete Restrict Rule"* error if your deletion or change would have deleted one or more rows in dependent tables, but the delete rule is RESTRICT. These are referential integrity (RI) constraint errors.

The error information panel for this error shows the names of the parent and dependent tables in the relationship that caused the error, and the name of the relationship. At the bottom of the panel are the names of the columns that comprise the key in both the parent and dependent tables, along with the value that caused the error.

You can enter the RE prefix command against the row, or enter the REDIT command from the error information panel, to edit the dependent table of the relationship and correct the error.

Related tasks

Editing related tables on page 192

Related references

REDIT primary command on page 992

"Check Constraint" errors

ZDT/Db2 reports a *"Check Constraint Error"* if a change violates a check constraint defined on either a column or the table. The panel shows the name of the constraint that caused the change to fail. You can press the ShowCon function key (F5), or issue the SHOWCON command, to display an ISPF Edit session showing the SQL text of the constraint.

"Row Not Found" errors

ZDT/Db2 reports a *"Row Not Found Error"* if the original row was not found when either a DELETE or UPDATE operation was attempted. This can occur because another user has changed the row since the data was originally retrieved from Db2®. The recommended action in this situation is to restart the Edit session with a current copy of the data. This can also occur if the Db2® object being edited has a TIME data type column and the system options are set to display the data for this column in USA format.

Related tasks

ZDT/Db2 system options (option 0.2) on page 65
DATE and TIME columns on page 168

Related references

Systems Options (1 of 4) panel on page 857

Deadlocks or timeouts

Db2® might encounter a deadlock or timeout condition (SQLCODE -911) if it attempts to implement changes made in a Z Data Tools/Db2 Edit session. If this occurs there is a delay, followed by the display of the SQL error pop-up panel with the explanation for the -911 SQLCODE.

If you encounter this error, be aware that:

- Changes made to any row marked with -911 have **not** been completed.
- Db2® has rolled back the current unit of work. This means that all uncommitted changes made in the current Edit session have been lost.
- After the rollback of the current unit of work, the data shown in the ZDT/Db2 Edit session might not reflect the current state of the data in the Db2® table.

The recommended course of action is to cancel the current Edit session and restart with a new copy of the data. If the problem persists contact your Db2® systems administrator for assistance in determining which other Db2® task is preventing access to the Db2® object.

Related tasks

Ending a Z Data Tools/Db2 editor session on page 133

Example of correcting an error during an Edit session

If an error occurs during an Edit session, ZDT/Db2 displays *"=ERR "*, *"=ERR R"*, or the SQLCODE in the prefix area of the row (or rows) in error. Figure 57: Example: error indicated during Edit session on page 206 shows an error has occurred for one row.

Figure 57. Example: error indicated during Edit session

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Table Edit                          1 of 42
TABLE HFMUSER.EMP                                                  Format TABL
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+---->
000000 ****  Top of data  ****
000001 000010 CHRISTINE<   I       HAAS<           A00      3978    01/01/1965
000002 000020 MICHAEL<     L       THOMPSON<       B01      3476    10/10/1973
000003 000030 SALLY<       A       KWAN<           C01      4738    04/05/1975
000004 000050 JAN<         B       GEYER<          E01      6789    08/17/1949
-803   000050 IRVING<      F       STERN<          D11      6423    09/14/1973
000006 000070 EVA<         D       PULASKI<        D21      7831    09/30/1980
000007 000090 EILEEN<      W       HENDERSON<      E11      5498    08/15/1970
000008 000100 THEODORE<    Q       SPENCER<        E21      0972    06/19/1980
000009 000110 VINCENZO<    G       LUCCHESI<       A00      3490    05/16/1958
000010 000120 SEAN<                O'CONNELL<      A00      2167    12/05/1963
000011 000130 DOLORES<     M       QUINTANA<       C01      4578    07/28/1971
000012 000140 HEATHER<     A       NICHOLLS<       C01      1793    12/15/1976
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

To determine the type of error, type `E` in the prefix area for the row in error and press Enter. ZDT/Db2 displays an error information panel that describes the error and instructions about what you can do to correct the error.

Figure 58. Example: error information describing the type of error that occurred during an Edit session

```
                  ┌──────────────────── Db2 Save Error Action ────────────────────┐
s                 │                                                               │ ──────────────
F                 │ Db2 reported an error while attempting to save this row.      │        1 of 42
L                 │                                                               │    Format TABL
                  │ Duplicate Row Error                                           │ ENO HIREDATE
                  │                                                               │     #7
                  │ The insert or update operation on this line would have        │ )   DATE
                  │ resulted in two rows with the same index key, and the         │     <---+---->
0                 │ index is defined as unique.                                   │
0                 │                                                               │     01/01/1965
0                 │ Instructions:                                                 │     10/10/1973
0                 │                                                               │     04/05/1975
0                 │    Press ENTER or enter EXIT to return to the edit            │     08/17/1949
E                 │    session and correct the error in this line.                │     09/14/1973
0                 │                                                               │     09/30/1980
0                 │    Enter CANCEL to terminate the Edit session.                │     08/15/1970
0                 │    Any changes made since the last commit point               │     06/19/1980
0                 │    will be lost.                                              │     05/16/1958
0                 │                                                               │     12/05/1963
0                 │                                                               │     07/28/1971
0                 │ Command ===>                                                  │     12/15/1976
C                 │  F1=Help       F2=Split      F3=Exit       F7=Backward      │    Scroll PAGE
                  │  F8=Forward    F9=Swap      F12=Cancel                       │   F6=RChange
                  └───────────────────────────────────────────────────────────────┘  F12=Cancel
```

You can then take one of the following actions:

- Cancel the Edit session by pressing the Cancel function key (F12). Any uncommitted changes are lost.
- Press Enter or the Exit function key (F3) to return to the Edit session.

- Use the REDIT primary command (if the error information panel offers it as an option) to edit the related table to correct the error.

  Note: If the error information panel does *not* offer the REDIT primary command as an option, REDIT displays an Edit session.

In the example shown in Figure 57: Example: error indicated during Edit session on page 206, the error has been caused by two rows having the same index key (rows 4 and 5 both have an EMPNO of 000050). To correct the error:

1. Press Enter to return to the Edit session.
2. Change the offending index key for row 5 to a unique value such as 000060.
3. Press the Exit function key (F3) to return to the Edit session.

Related references

Table Edit panel on page 868
REDIT primary command on page 992

Chapter 6. Creating and dropping Db2® objects

You can use the Objects utility to create and drop Db2® objects without needing to know the SQL commands for these operations.

The Db2® Object Functions panel is option 3.2 from the Primary Option Menu panel. From the Db2® Object Functions panel you can select the Db2® object type you want to create or drop:

1. Specify the relevant number in the **Object Type** entry field. When you first display the Db2® Object Functions panel, this field is initialized to 3 (Table).
2. If you intend to drop an object but would like ZDT/Db2 to display a Drop Confirmation panel before it actually drops the object you have specified, ensure the **Confirm object drop** option is checked.
3. To *create* the specified object type, type `C` on the command line.

   To *drop* the specified object type, type `D` on the command line.
4. Press Enter.

Table 9: Supported actions for each Db2 object on page 208 shows the Db2® object types you can create or drop.

Table 9. Supported actions for each Db2® object

Object Type	Create	Drop
Database	Yes	Yes
Table space	Yes	Yes
Table	Yes	Yes
View	Yes	Yes
Alias	Yes	Yes
Index	Yes	Yes
Synonym	Yes	Yes
Distinct Type	Yes	Yes
Function	Yes	Yes
Stored Procedure	Yes	Yes
Trigger	Yes	Yes
Auxiliary Table	Yes	No[14]

14. **Notes:**

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Db2 Object Functions panel on page 622
Drop Confirmation panel on page 640

Working with the Create and Drop panels

Here are a few general points relating to the usage of the Create and Drop panels:

You cannot use wildcard characters when using the Create or Drop panels.

You can type **SQL** on the command line of the main Create and Drop panels and the Drop Confirmation panel to examine the SQL that would be constructed given the current panel inputs.

If you receive a validation error when inputting data on the panels, a brief error message is displayed on the upper right of the panel. To obtain a more detailed explanation of the error message, press the Help function key (F1). Also, if you receive an SQL error during the Create or Drop operation, you can press the Help function key (F1) to receive a more detailed explanation. Press the Help function key (F1) again to examine the SQL statement and the cause of the error.

Some of the Create panels such as the Create Table Space panel and the Create Function panel incorporate a number of subsidiary panels to collect the Create statement inputs. Generally, only a few of these panels are required, therefore you do not need to visit all of the lower-level panels before executing the Create statement. The inputs and the panels that are required are indicated on the main panel. Generally, you can visit and revisit the subsidiary panels in any order but exceptions are indicated on the main panel. If you receive an SQL error during the create operation, you need only return to the panels that are in error to correct the errors and then try the Create operation again. When inputting your data on the subsidiary panel, you can press Enter to validate your changes, the Exit function key (F3) to save your changes and return to the main panel, or the Cancel function key (F12) to return to the main panel without saving your changes.

Many of the subsidiary panels use ISPF tables to manage object information that can have more than one occurrence, such as the columns of a table or the parameters of a function. Each line in the table contains a set of values for an occurrence of the piece of information.

When editing the ISPF tables, you can repeat, delete, and insert lines when the CMD field is displayed using the following commands:

1. Use table drop. Populated auxiliary tables can only be dropped by dropping their associated base table. To drop an unpopulated auxiliary table without dropping its base table:
   - Use the drop table function (by selecting Object Type 3 and entering D on the command line on the Db2® Object Functions panel.
   - Specify the auxiliary table as the table to drop.

I

Inserts a new (blank) line at the current position.

I*(n)*

Inserts *n* new (blank) lines at the current position.

D

Deletes the line.

D*(n)*

Deletes *n* lines, starting at the current position.

R

Repeats the line.

R*(n)*

Repeats the line *n* times.

If the **Cmd** field is not displayed, you can only edit some fields in the existing lines.

SQL Keywords

Table 10: Create Database on page 211 through to Table 21: Create Auxiliary Table on page 223 show, for each of the Create panels (and, where applicable, for each subsidiary panel), the SQL keyword (or variable) that relates to each entry field on that panel.

For example, Table 10: Create Database on page 211 shows that, on the main Create Database panel, the optional entry field **Storage group name** relates to the SQL keyword STOGROUP. When ZDT/Db2 builds the CREATE DATABASE statement, if you have entered a name (*stogroup-name*) in the **Storage group name** entry field, ZDT/Db2 generates the storage group part of the CREATE DATABASE statement syntax as:

```
STOGROUP stogroup-name
```

Note:

1. In some cases, a keyword that is valid for the connected Db2® version is not displayed on any panel because, although the keyword is supported, it is not required to build the SQL statement. The create index **TYPE** parameter is an example of this.
2. Some keyword values are managed entirely by ZDT/Db2 and are displayed only as output fields on the panels, and are included here for reference. For example, for the **PART** keyword, ZDT/Db2 generates the value set

for **NUMPARTS**. Fields that are displayed elsewhere as input fields are not redocumented if they reappear as output fields.

3. Where possible, fields are associated with SQL keywords (rather than the values associated with those keywords). Otherwise, they are associated with the value-names shown in the relevant SQL syntax diagram. Occasionally more detailed context is given to avoid ambiguity.

Table 10. Create Database

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Database panel on page 529.)	
	(Database) Name	*database-name*
	Storage group name	**STOGROUP**
	Buffer pool for table spaces	**BUFFERPOOL**
	Buffer pool for indexes	**INDEXBP**
	Database usage	**AS WORKFILE**
	Database usage	**AS TEMP**
	Data encoding	**CCSID**
	Member name	**FOR MEMBER**

Table 11. Create Table Space

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Table Space panel on page 564.)	
	(Table space) Name	*table-space-name*
	Database	**IN**
	Buffer pool	**BUFFERPOOL**
1. Type	(See Create Table Space: Type panel on page 573.)	
	Table Space Type	**LARGE**
	Table Space Type	**LOB**
	Segment size	**SEGSIZE**

Table 11. Create Table Space

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	No. of partitions	**NUMPARTS**
	Max. partition size	**DSSIZE**
2. Allocation	(See Create Table Space: Allocation panel on page 566.)	
	VCAT name	**VCAT**
	Storage group name	**STOGROUP**
	Primary space	**PRIQTY**
	Secondary space	**SECQTY**
	Free pages	**FREEPAGE**
	Free space	**PCTFREE**
	Maximum rows per page	**MAXROWS**
3. Data Storage Options	(See Create Table Space: Data Storage Options panel on page 568.)	
	Use data compression	**COMPRESS**
	Close if not in use	**CLOSE**
	Erase data on delete	**ERASE**
	Define data set now	**DEFINE**
	INSERT uses clustering index	**MEMBER CLUSTER**
	Track modified changes	**TRACKMOD**
	Log changes to LOB columns	**LOG**
	Data encoding	**CCSID**
4. Locking Options	(See Create Table Space: Locking Options panel on page 572.)	
	Locksize parameter	**LOCKSIZE**
	Application Locking	**LOCKMAX**
5. Data Sharing Options	(See Create Table Space: Data Sharing Options panel on page 567.)	
	Selective Partition Locking	**LOCKPART**
	Group Buffer Pool Usage	**GBPCACHE**
6. Define Partitions	(See Create Table Space: Define Partitions panel on page 570.)	

Table 11. Create Table Space

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Pt. No.	**PART**
	VCAT name	**VCAT**
	Storage group name	**STOGROUP**
	Primary space	**PRIQTY**
	Secndry space	**SECQTY**
	Erase Data	**ERASE**
	Free pages	**FREEPAGE**
	Free space	**PCTFREE**
	Use Data Cmpr	**COMPRESS**
	Group BP Caching	**GBPCACHE**

Table 12. Create Table

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Table panel on page 561.)	
	(New Table) Owner	*table-name*
	(New Table) Name	*table-name*
	(New Table) Database	**IN**
	(New Table) Table Space	**IN**
	(Model Table) Owner	**LIKE**
	(Model Table) Name	**LIKE**
1. Columns	(See Create Table: Columns panel on page 578.)	
	Column Name	*column-name*
	Data Type	*built-in-data-type*
	Len	*built-in-data-type*
	Pr	*built-in-data-type* (numeric only)

Table 12. Create Table

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Sc	*built-in-data-type* (decimal only)
	For Data	**FOR … DATA**
	User Defined Data Type	*distinct-type-name*
2. Default Values	(See Create Table: Nulls and Default Values panel on page 584.)	
	Not Null	**NOT NULL**
	Default Value	**WITH DEFAULT**
3. Options	(See Create Table: Options panel on page 587.)	
	Editproc	**EDITPROC**
	Validproc	**VALIDPROC**
	OBID value	**OBID**
	Audit Options	**AUDIT**
	Data Storage Options	**CCSID**
	Log data capture changes	**DATA CAPTURE**
	Restrict drop of table	**WITH RESTRICT ON DROP**
4. Primary Key	(See Create Table: Unique Constraints panel on page 595.)	
	Constraint Name	**CONSTRAINT** *constraint-name* **PRIMARY KEY**
	Order	**PRIMARY KEY(***column-name …* **)**
5. Foreign Key	(See Create Table: Column Referential Constraints panel on page 576.)	
	Constrnt Name	**FOREIGN KEY**
	Owner	**REFERENCES**
	Table Name	**REFERENCES**
	Column Name	**REFERENCES**
	ON DLT	**ON DELETE**
6. Check Constraints	(See Create Table: Column Check Constraints panel on page 575.)	
	Table Constraint	**CONSTRAINT** *constraint-name* **CHECK**
	Check Condition	**CHECK(***check-condition …* **)**

Table 12. Create Table

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	(Constraint) Name	**CONSTRAINT** *constraint-name* **CHECK**
	(Constraint) Condition	**CHECK(***check-condition ...* **)**
7. Procedure Exits	(See Create Table: Procedure Exits panel on page 589.)	
	Name	**FIELDPROC**
	Parameters	**FIELDPROC**
8. Generate Values	(See Create Table: Generate Values panel on page 580.)	
	Gen (A/D)	**GENERATE**
	As Idty (Y/N)	**AS IDENTITY**
	Start value	**START WITH**
	Incr value	**INCREMENT BY**
	Caching Option	**(NO) CACHE**

Table 13. Create View

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create View panel on page 601.)	
	Owner	*view-name*
	Name	*view-name*
	Select statement	**AS**
	Column names	*column-name ...*
	With Check Option	**WITH ... CHECK OPTION**

Table 14. Create Alias

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Alias panel on page 527.)	

Table 14. Create Alias

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	(Alias) Owner	*alias-name*
	(Alias) Name	*alias-name*
	(Table or View) Location	**FOR**
	(Table or View) Owner	**FOR**
	(Table or View) Name	**FOR**

Table 15. Create Index

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Index panel on page 542.)	
	Owner	*index-name*
	Name	*index-name*
	Table Owner	**ON**
	Table Name	**ON**
	Buffer Pool	**BUFFERPOOL**
1. Index Type	(See Create Index: Type panel on page 552.)	
	Uniqueness	**UNIQUE WHERE NOT NULL**
	Internal format	**TYPE**
	Clustering Index Option	**CLUSTER**
	Piecesize	**PIECESIZE**
	Multiplier	**PIECESIZE**
2. Column Selection	(See Create Index: Column Selection panel on page 546.)	
	Order	*column-name*
	(A/D)	*column-name*
3. Allocation	(See Create Index: Allocation panel on page 545.)	
	VCAT name	**VCAT**

Table 15. Create Index

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Storage group name	**STOGROUP**
	Primary space	**PRIQTY**
	Secondary space	**SECQTY**
	Free pages	**FREEPAGE**
	Free space	**PCTFREE**
4. Options	(See Create Index: Options panel on page 547.)	
	Close if not in use	**CLOSE**
	Erase data on delete	**ERASE**
	Defer building index	**DEFER**
	Define data set now	**DEFINE**
	Allow COPY of index	**COPY**
	Group Buffer Pool Usage	**GBPCACHE**
5. Partitions	(See Create Index: Partitions panel on page 549.)	
	Parttn. Number	**PART**
	VCAT name	**VCAT**
	Storage group name	**STOGROUP**
	Primary space	**PRIQTY**
	Secndry space	**SECQTY**
	Erase Data	**ERASE**
	Free pages	**FREEPAGE**
	Free space	**PCTFREE**
	Group BP Caching	**GBPCACHE**
6. Partition Values	(See Create Index: Partition Values panel on page 550.)	
	Value	**VALUES**

Table 16. Create Synonym

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Synonym panel on page 560.)	
	(Synonym) Name	*synonym*
	(Table) Owner	**FOR**
	(Table) Name	**FOR**

Table 17. Create Distinct Type

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Distinct Type panel on page 531.)	
	Schema	*distinct-type-name*
	Name	*distinct-type-name*
	Source Type	**AS**
	Length	**AS**
	Scale	**AS**
	For Data	**FOR … DATA**
	Encoding Method	**CCSID**

Table 18. Create Function

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Function panel on page 536.)	
	Schema	*function-name*
	Name	*function-name*
	Unique Name	**SPECIFIC**
1. Parameters	(See Create Function: Parameters panel on page 539.)	
	Parameter Name	*parameter-name*

Table 18. Create Function

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Data Type	*built-in-data-type*
	Len	*built-in-data-type*
	Pr	*built-in-data-type*
	Sc	*built-in-data-type*
	For Data	**FOR … DATA**
	User Defined Data Type	*distinct-type-name*
2. Parameter Types	(See Create Function: Parameter Types panel on page 537.)	
	As LOC	**AS LOCATOR**
	Owner	**TABLE LIKE**
	Name	**TABLE LIKE**
3. Returned Data Type	(See Create Function: Returned Data Type panel on page 540.)	
	(Returns) Data Type	**RETURNS**
	(Returns) Schema	**RETURNS**
	(Returns) Length	**RETURNS**
	(Returns) Scale	**RETURNS**
	(Returns) For Data	**RETURNS … FOR … DATA**
	(Returns) Encoding Method	**RETURNS … CCSID**
	(Cast From) Data Type	**CAST FROM**
	(Cast From) Length	**CAST FROM**
	(Cast From) Scale	**CAST FROM**
	(Cast From) For Data	**CAST FROM … FOR … DATA**
	(Cast From) Encoding Method	**CAST FROM … CCSID**
	As Locator	**RETURNS … AS LOCATOR**
4. Option List (1/2)	(See Create Function: Option List (1/2) panel on page 533.)	
	External Name	**EXTERNAL NAME**
	Scratchpad	**(NO) SCRATCHPAD**

Table 18. Create Function

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Package Collection	**(NO) COLLID**
	Workload Manager Environment	**WLM ENVIRONMENT**
	Time Limit	**ASUTIME (NO) LIMIT**
	Language	**LANGUAGE**
	SQL	**(READS)(NO)(MODIFIES)** **(CONTAINS) SQL (DATA)**
	Security	**SECURITY**
	Parallel	**(DIS)ALLOW PARALLEL**
	Run-time Options	**RUN OPTIONS**
5. Option List (2/2)	(See Create Function: Option List (2/2) panel on page 534.)	
	Main Program	**PROGRAM TYPE**
	Stay Resident	**STAY RESIDENT**
	Deterministic	**(NOT) DETERMINISTIC**
	Called on null input	**(RETURNS NULL)(CALLED) ON NULL INPUT**
	No external action	**(NO) EXTERNAL ACTION**
	Final Call	**(NO) FINAL CALL**
	DBINFO argument passed	**(NO) DBINFO**

Table 19. Create Procedure

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Procedure panel on page 553.)	
	Schema	*procedure-name*
	Name	*procedure-name*
	Result Sets	**DYNAMIC RESULT SET(S)**
1. Parameters	(See Create Procedure: Parameters panel on page 558.)	
	Parameter Name	*parameter-name*

Table 19. Create Procedure

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Data Type	*built-in-data-type*
	Len	*built-in-data-type*
	Pr	*built-in-data-type*
	Sc	*built-in-data-type*
	For Data	**FOR … DATA**
	User Defined Data Type	*distinct-type-name*
2. Parameter Types	(See Create Procedure: Parameter Types panel on page 557.)	
	As LOC	**AS LOCATOR**
	Owner	**TABLE LIKE**
	Name	**TABLE LIKE**
	IN OUT INOUT	**(IN)(OUT)(INOUT)**
4. Option List (1/2)	(See Create Procedure: Option List (1/2) panel on page 554.)	
	External Name	**EXTERNAL NAME**
	Package Collection	**(NO) COLLID**
	Workload Manager Environment	**(NO) WLM ENVIRONMENT**
	Time Limit	**ASUTIME (NO) LIMIT**
	Language	**LANGUAGE**
	SQL	**(READS)(NO)(MODIFIES)** **(CONTAINS) SQL (DATA)**
	Security	**SECURITY**
	Parameter Style	**PARAMETER STYLE**
	Run-time Options	**RUN OPTIONS**
5. Option List (2/2)	(See Create Procedure: Option List (2/2) panel on page 556.)	
	Main Program	**PROGRAM TYPE**
	Stay Resident	**STAY RESIDENT**
	Deterministic	**(NOT) DETERMINISTIC**
	DBINFO argument passed	**(NO) DBINFO**

Table 19. Create Procedure

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Commit on return	**COMMIT ON RETURN**

Table 20. Create Trigger

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Trigger panel on page 596.)	
	Schema	*trigger-name*
	Name	*trigger-name*
1. Type	(See Create Trigger: Details panel on page 598.)	
	(Order) No cascade before	**NO CASCADE BEFORE**
	(Order) After	**AFTER**
	(Type) Insert	**INSERT**
	(Type) Delete	**DELETE**
	(Type) Update	**UPDATE**
	Columns	**UPDATE OF**
	(Trigger Table) Name	**ON**
	(Trigger Table) Owner	**ON**
	Old correlation name	**OLD**
	New correlation name	**NEW**
	Old table identifier	**OLD TABLE**
	New table identifier	**NEW TABLE**
	(Executed) For each modified row	**FOR EACH ROW**
	(Executed) Once	**FOR EACH STATEMENT**
2. Search Condition	(See Create Trigger: Search Condition panel on page 600.)	
	Search Condition	**WHEN**
3. SQL Statement	(See Create Trigger: SQL statement panel on page 600.)	

Table 20. Create Trigger

(continued)

Main/subsidiary panel	Field	SQL Keyword(s)
	Triggered SQL Statement(s)	**BEGIN ATOMIC**

Table 21. Create Auxiliary Table

Main/subsidiary panel	Field	SQL Keyword(s)
Main	(See Create Auxiliary Table panel on page 528.)	
	(Auxiliary Table) Owner	*aux-table-name*
	(Auxiliary Table) Name	*aux-table-name*
	(Table Space) Name	**IN**
	Database	**IN**
	(Base Table) Owner	**STORES**
	(Base Table) Name	**STORES**
	(Base Table Column) Name	**COLUMN**
	Partition	**PART**

Creating a database

To create a new database, perform the following on the Db2® Object Functions panel:

1. Type `1` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Database panel.

The Create Database panel defines a Db2® database at the current server.

The only required field on the panel is the first, where you enter the name for the new database. All of the other fields on the panel are optional, although values are often specified for the second group of input fields.

The database options and member selection sections of the panel contain input fields that you can use to define a work file or temporary database.

The default data encoding scheme for z/OS® systems is EBCDIC. You can specify ASCII encoding if required.

For workfile databases, you can also specify a member name (for use in a data sharing environment only).

Related references

Db2 Object Functions panel on page 622
Create Database panel on page 529

Creating a table space

To create a new table space, perform the following on the Db2® Object Functions panel:

1. Type `2` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Table Space panel.

Use the Create Table Space panel to define a simple, segmented or partitioned table space at the current server.

Related tasks

Table Space Type (Create Table Space) on page 224
Table Space Allocation (Create Table Space) on page 225
Data Storage (Create Table Space) on page 225
Locking (Create Table Space) on page 225
Data Sharing (Create Table Space) on page 225
Partitions (Create Table Space) on page 226

Related references

Db2 Object Functions panel on page 622
Create Table Space panel on page 564
Create Table Space: Type panel on page 573
Create Table Space: Allocation panel on page 566
Create Table Space: Data Storage Options panel on page 568
Create Table Space: Locking Options panel on page 572
Create Table Space: Data Sharing Options panel on page 567
Create Table Space: Define Partitions panel on page 570

Table Space Type (Create Table Space)

In Db2®, table space is either non-partitioned, or partitioned. A non-partitioned table space can be segmented (a partitioned table space cannot). Non-partitioned means that there is a 1:1 relationship between the table space and the data set defined to store the data. Partition means that many data sets are used to store the data.

To make more effective use of non-partitioned table spaces, by improving the support for multiple table definitions within the one table space, use segmented table spaces.

The maximum partition size for a partitioned table space determines the amount of data that can be stored within the table space. Large values imply the creation of multi-volume table spaces capable of storing GB of data.

Related references

Create Table Space: Type panel on page 573

Table Space Allocation (Create Table Space)

The Create Table Space: Allocation panel is optional, although most users need to specify a Db2® storage group and the size of the table space.

You have the option of defining Db2® data sets externally, or Db2® can create the data sets on behalf of you. The former option is selected by specifying a VCAT name in the data set parameters section of the panel. The most common option is to specify a (Db2®) storage group name. These parameters are mutually exclusive.

Related references

Create Table Space: Allocation panel on page 566

Data Storage (Create Table Space)

Use the Create Table Space: Data Sharing Options panel to specify options that control various aspects of data management for the function.

Related references

Create Table Space: Data Sharing Options panel on page 567

Locking (Create Table Space)

The locksize parameter specifies the size of locks within the table space.

You can select from six different locking parameters. Page and Row cannot be specified for LOB table spaces. Table can only be specified for a segmented table space.

You can also specify a maximum value for the number of locks that application programs can hold on a table space before lock escalation occurs.

Related references

Create Table Space: Locking Options panel on page 572

Data Sharing (Create Table Space)

The Create Table Space: Data Sharing Options panel is optional and contains options that control how the table space behaves in a data sharing environment. In most cases, the default values are appropriate.

Related references

Create Table Space: Data Sharing Options panel on page 567

Partitions (Create Table Space)

The Create Table Space: Define Partitions panel is optional, even for the definition of a partitioned table space. However, you might want to specify parameters for each partition of a partitioned table space, rather than relying on the values set at the table space level.

For a partitioned table space, there is an optional 'USING CLAUSE' for every partition of the table space. You can define the number of partitions on the Table Space Type panel before selecting this panel. The table at the bottom of the panel is pre-formatted with the required number of rows (one for each partition).

Alternatively, you can select the Define Partitions panels directly. The number of pre-formatted rows reflect the value specified for a number of partitions (the default value is 1). You can then adjust the values for this single partition to what is required. After this is done, you can use the R prefix command (optionally followed by an integer) to expand the number of partitions to the required number. The value specified for number of partitions on the Table space Type panel is updated accordingly.

Note: The individual partitions of a partitioned table space are typically defined with similar parameters. It is easier for you to define the values once, then repeat them, rather than type the same information in for each row repeatedly.

The Db2® default values for Free Pages, Percent Free, Data Compression and GBPCACHE are used.

Related references

Create Table Space: Define Partitions panel on page 570
Create Table Space: Type panel on page 573

Creating a table

To create a new table, perform the following on the Db2® Object Functions panel:

1. Type `3` in the **Object Type** entry field.
2. Type `C` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Table panel.

Use the Create Table utility to create a new Db2® table. If a table already exists that is similar to the one you want to create, you can use that table as a model for the new table.

Related tasks

1. Columns (Create Table) on page 231
2. Nulls and default values (Create Table) on page 231

3. Options (Create Table) on page 232
4. Unique constraints (Create Table) on page 232
5. Referential constraints (Create Table) on page 232
6. Check constraints (Create Table) on page 234
7. Procedure exits (Create Table) on page 235
8. Generate values (Create Table) on page 235

Related references

Db2 Object Functions panel on page 622
Create Table panel on page 561
Create Table: Columns panel on page 578
Create Table: Nulls and Default Values panel on page 584
Create Table: Options panel on page 587
Create Table: Unique Constraints panel on page 595
Create Table: Column Referential Constraints panel on page 576
Create Table: Column Check Constraints panel on page 575
Create Table: Procedure Exits panel on page 589
Create Table: Generate Values panel on page 580

Creating a table using a model

The Create Table utility lets you use an existing table or view as a model for the new table you want to create. You can use this modeling facility to create a table in either of these ways:

- With columns that have exactly the same name and description as the model table or view: Usage option 1.

  Note: If you use Usage option 1, none of the Table Creation options related to columns are available (Db2® copies the column information about the model table or view when it issues the CREATE TABLE statement). However, you can still specify non-column details on the main panel and by selecting Table Creation option 3 (**Table options**).

- Based on the model table or view but where you want to change some of the column details : Usage option 2, Table Creation options 1–8.

To create a table with columns that have exactly the same name and description as the model table or view:

1. Specify the name of the table you want to create in **New Table**.
2. Specify the name of the model table or view in **Model Table/View**. You can optionally qualify the name of the model table or view by specifying an owner. If you do not specify a model table (or view) owner, ZDT/Db2 uses your current SQL ID.
3. Select Usage option 1 (**Generate LIKE clause**).
4. Select the Table Creation option **Create the table**.

5. Press Enter.

   ZDT/Db2 generates the following SQL statement:

   ```
   CREATE TABLE new_table_name LIKE model_table_name
   ```

   before sending it to Db2® for execution. ZDT/Db2 issues a message to confirm that the generated SQL statement has been run. For further details about the LIKE clause in the CREATE TABLE statement, see the *Db2 UDB for z/OS SQL Reference*.

   Note: To view the generated SQL statement, either before or after it is issued, enter `SQL` on the command line.

To create a table based on the model table or view, but where you want to change some of the column details:

1. Specify the name of the Db2® table you want to create in **New Table**.
2. Specify the name of the model Db2® table or view in **Model Table/View**. You can optionally qualify the name of the model table or view by specifying an owner. If you do not specify a model table (or view) owner, ZDT/Db2 uses your current SQL ID.
3. Select Usage option 2 (**Load table information**).
4. Press Enter.

   ZDT/Db2 displays the Create Table Model Load panel.

   Note:
   The Create Table Model Load panel is displayed if you:
   - Initially specify, or subsequently change, the **Model Table/View** details,
   - Select Usage option 2 (**Load table information**), and
   - Press Enter.

   Use the Create Table Model Load panel to selectively load information describing the model table or view from the Db2® catalog into the ISPF variables and tables used to generate the CREATE TABLE statement. You can also use this panel to specify whether you want this information to replace the current information, or to be added to it.

5. Select the **Model Load Options** you want by typing a slash (/) next to the required option. (ZDT/Db2 selects the first three options for you.) To deselect a model load option, type a blank in place of the slash.

   Note: If no data exists in the model table or view for any of the model load options you select, ZDT/Db2 issues the message *"Some data not loaded"* when it redisplays the main panel.

   When you return to the main Create Table panel, you can change or add to this information by selecting the appropriate table creation option 1–8. For details, see Table Creation options on page 230.

Note: If you use a view as the model, ZDT/Db2 loads only the database name, table space name, encoding scheme, and column information.

6. To specify whether you want ZDT/Db2 to use the information about the model table or view as a refresh, or if you want ZDT/Db2 to add it to the current information for the table you are creating, select the appropriate **Refresh/Add** option.
7. If the model table has referential constraints, specify whether you want single-column referential constraints loaded as *column* referential constraints or *table* referential constraints by selecting one of the following options:
    - 1. Column referential constraints
    - 2. Table referential constraints

Note:

a. ZDT/Db2 does not check for duplicate column names if multiple model tables (or views) are loaded or model table (or view) columns are added to manually-defined columns.
b. If you select any of the key or constraint options (**Primary key information**, **Unique key information**, **Referential constraints** or **Check constraints**), the **Column information** option is selected automatically.
c. ZDT/Db2 loads all check constraint information in the form of *table* check constraints.

8. Press Enter.

   ZDT/Db2 loads information into the ISPF variables and tables used to generate the CREATE TABLE statement according to the **Model Load Options** and **Refresh/Add** options you have selected and returns you to the main Create Table panel.

   Note: To view the generated SQL statement, type `SQL` on the command line of the main Create Table panel.

9. To specify further information for the table you want to create, select any of the table creation options 1–8 and press Enter.

   ZDT/Db2 validates your input and displays a panel that you use to enter information for the table creation option you have selected. For details about each table creation option, see Table Creation options on page 230.

10. Select the Table Creation option **Create the table** on the main panel.
11. Press Enter.

    ZDT/Db2 generates the following SQL statement:

```
CREATE TABLE new_table_name other_details
```

where *other_details* represents SQL clauses based on the details you have selected using the model table or view and Table Creation options. ZDT/Db2 sends the generated SQL statement to Db2® and issues a message to confirm that it has been successfully run.

Note: To view the generated SQL statement, either before or after it is issued, enter `SQL` on the command line.

Related references

Create Table: Model Load panel on page 582
Create Table panel on page 561

Creating a table without using a model

To create a new table without using a model (for instance, if no table already exists that is similar to the one you want to create):

1. Specify the name of the Db2® table you want to create in **New Table**.
2. Leave the owner and name of the model Db2® table in **Model Table** blank.
3. Select any of the Table Creation options 1–8 and press Enter. The default setting is Option 1 (**Columns**).

   Note: Before you can specify any of the table creation options 2, 4, 5, 6, 7, or 8, you must create at least some of the columns for the new table (using option 1).

   ZDT/Db2 validates your input and displays a panel that you use to enter information for the table creation option you have selected.

   For information about each table creation option, see Table Creation options on page 230.
4. Select the Table Creation option **Create the table**
5. Press Enter.

   ZDT/Db2 generates the following SQL statement:

   ```
   CREATE TABLE new_table_name other_details
   ```

   where *other_details* represents SQL clauses based on the details you have selected using the Table Creation options. ZDT/Db2 sends the generated SQL statement to Db2® and issues a message to confirm that it has been successfully run.

   Note: To view the generated SQL statement, either before or after it is issued, enter `SQL` on the command line.

Table Creation options

Use the table creation options on the main Create Table panel to provide additional information to define the table you are creating.

If you are using a model table with Usage option 2, you can select any of the options 1–8 before using the Table Creation option **Create the table** to create the table. If you have already used the Model Load panel to provide information for the table you want to create, use the table creation options to supplement or change details you have already selected.

If you are *not* using a model table, you must select option 1 (**Columns**) and then optionally options 2–8, before using the Table Creation option **Create the table** to create the table.

You can revisit each of the table creation options as many times as you want to add or modify the data.

If you select one of the table creation options 1 to 8, ZDT/Db2 displays a panel that you use to enter information for the creation option you have selected. If you select the Table Creation option **Create the table**, ZDT/Db2 generates and issues a CREATE TABLE SQL statement.

The following pages describe the panels for each creation option.

Related references

Create Table panel on page 561
Create Table: Model Load panel on page 582

1. Columns (Create Table)

The Create Table Columns panel is displayed if you select Table Creation option 1 (**Columns**) on the main Create Table panel. You use this panel to define columns for the table you are creating, or to change information for columns you have already defined.

The lower section of the panel shows the columns defined for the table. (If you have not yet defined any columns, this part of the panel shows a single line of empty entry fields.) If you have used the modeling facility, the panel shows the column details for each column in the model table.

Related references

Create Table: Columns panel on page 578
Create Table panel on page 561

2. Nulls and default values (Create Table)

The Nulls and Default Values panel is displayed if you select Table Creation option 2 (**Nulls/default values**) on the main Create Table panel.

The Nulls and Default Values panel shows the columns defined for the table. If you have used the modeling facility, the panel shows the column details for each column in the model table. You use this panel to specify information to define the null attributes and default values for the columns of the table you are creating.

Related references

Create Table: Nulls and Default Values panel on page 584
Create Table panel on page 561

3. Options (Create Table)

The Options panel is displayed if you select Table Creation option 3 (**Table options**) on the main Create Table panel.

You use the Options panel to specify information to define options for the table you are creating.

Related references

Create Table: Options panel on page 587
Create Table panel on page 561

4. Unique constraints (Create Table)

The Unique Constraints panel is displayed if you select Table Creation option 4 (**Unique Constraints**) on the main Create Table panel.

You use the Unique Constraints panel to specify information to define a primary key and up to three additional unique keys for the table you are creating. The panel shows only those columns that are defined as NOT NULL (and therefore eligible to be selected as part of either the primary key or a unique key).

To define the primary key or any of the three unique keys, enter s or a number in one of the **Column Order in Key** fields to add a column to the key. If you enter s, ZDT/Db2 generates the next highest number for that key. To deselect a column that is already selected, clear the displayed number by overtyping it with spaces.

Note: The actual numbers are not important; ZDT/Db2 uses their relative order to define the column order for the key.

You can optionally specify constraint names for the primary key and the three unique keys.

Related references

Create Table: Unique Constraints panel on page 595
Create Table panel on page 561

5. Referential constraints (Create Table)

If you select Table Creation option 5 (**Referential Constraints**) on the main Create Table panel, ZDT/Db2 displays either the Column Referential Constraints panel or the Table Referential Constraints panel. To change from one panel to the other, press the F11 function key.

Note: The first time you select the **Referential Constraints** creation option, ZDT/Db2 displays the Column Referential Constraints panel. Thereafter, within the same invocation of the Create Table utility, each time you select the **Referential Constraints** creation option, ZDT/Db2 displays whichever panel (column or table) it last displayed.

Column referential constraints and table referential constraints

You can specify referential constraints associated with individual columns (column referential constraints), or with the entire table (table referential constraints).

- A *column referential constraint* relates a column (the foreign key) in the table being created (the dependent table) to an equivalent column (the parent key) in another table (the parent table), and restricts non-null values of the foreign key column to the values of the parent key column.

- A *table referential constraint* relates a set of columns (the foreign key) in the table being created (the dependent table) to an equivalent set of columns (the parent key) in another table (the parent table), and restricts non-null values of the foreign key to the values of the parent key.

Note:

1. The parent key must be a primary or unique key on the parent table.
2. If you do not specify the parent key column name (or names), the foreign key is related to the primary key of the parent table.

To specify column referential constraints, use the Column Referential Constraints panel.

The Column Referential Constraints panel shows the columns defined for the table you are creating. If you have used the modeling facility, the panel shows the column details for each column in the model table. You use this panel to specify information to define column referential constraints for the table you are creating. To define a referential constraint, you must at least specify the parent table name. To indicate a column does not have a referential constraint, leave all the fields for that column blank.

To specify table referential constraints, use the Table Referential Constraints panel. Each row of the panel defines a single table referential constraint.

You use the Table Referential Constraints panel to specify information to define table referential constraints for the table you are creating. To define a referential constraint, you must at least specify the parent table name.

The Create Table Referential Constraint panel is displayed if you enter S in the **Cmd** entry field.

The Create Table Referential Constraint panel shows the columns defined for the table you are creating (the dependent table). If you have used the modeling facility, the panel shows the column details for each column in the model table. You use

this panel to select the dependent table columns (the foreign key) and specify the parent table columns (the parent key) to which they relate.

Related references

Create Table panel on page 561
Create Table: Column Referential Constraints panel on page 576
Create Table: Table Referential Constraint panel on page 592

6. Check constraints (Create Table)

If you select Table Creation option 6 (**Check Constraints**) on the main Create Table panel, ZDT/Db2 displays either the Column Check Constraints panel or the Table Check Constraints panel. To change from one panel to the other, press the F11 function key. You use these panels to specify information to define check constraints for the table you are creating.

Note: The first time you select the **Check Constraints** creation option, ZDT/Db2 displays the Column Check Constraints panel. Thereafter, within the same invocation of the Create Table utility, each time you select the **Check Constraints** creation option, ZDT/Db2 displays whichever panel (column or table) it last displayed.

Column check constraints and table check constraints

You can specify check constraints associated with individual columns (column check constraints), or with the entire table (table check constraints). Both types of check constraint produce the same result when the statement is issued, and you can specify any valid check condition as either type. However, you might prefer to use table check constraints for check conditions that would violate the SQL standard (not enforced by Db2®) that a column check constraint only refers to the column on which it is defined.

Note:

1. If model table information is loaded into the Create Table utility, all check constraints defined on the table are loaded as table check constraints.

To specify column check constraints, use the Column Check Constraints panel.

The Column Check Constraints panel shows the columns defined for the table you are creating. If you have used the modeling facility, the panel shows the column details for each column in the model table.

To specify table check constraints, use the Table Check Constraints panel. Each row of the panel defines a single table check constraint.

You use the Table Check Constraints panel to specify an SQL check condition for each check constraint you want to define for the table you are creating.

Related references

Create Table panel on page 561
Create Table: Column Check Constraints panel on page 575
Create Table: Table Check Constraints panel on page 590

7. Procedure exits (Create Table)

The Procedure Exits panel is displayed if you select Table Creation option 7 (**Procedure exits**) on the main Create Table panel.

You use the Procedure Exits panel to specify information to define procedure exits for the table you are creating.

Related references

Create Table: Procedure Exits panel on page 589
Create Table panel on page 561

8. Generate values (Create Table)

The Generate Values panel is displayed if you select Table Creation option 8 (**Generate value**) on the main Create Table panel.

You use the Generate Values panel to specify information to define generated values for identity or ROWID columns in the table you are creating.

Related references

Create Table: Generate Values panel on page 580
Create Table panel on page 561

Creating a view

To create a new view, perform the following on the Db2® Object Functions panel:

1. Type `4` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create View panel.

Use the Create View panel to create a new view at the current location.

Related references

Db2 Object Functions panel on page 622
Create View panel on page 601

Creating an alias

To create a new alias, perform the following on the Db2® Object Functions panel:

1. Type `5` in the **Object Type** entry field.
2. Type `C` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Alias panel.

Use the Create Alias panel to create an alias for a table or a view at the correct location.

An alias is an alternative name for a table or view. The table or view need not exist when the alias is created. The table or view can exist at a different location to the local Db2® system.

Related references

Db2 Object Functions panel on page 622
Create Alias panel on page 527

Creating an index

To create a new index, perform the following on the Db2® Object Functions panel:

1. Type `6` in the **Object Type** entry field.
2. Type `C` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Index panel.

Use the Create Index panel to create an index for a pre-existing base table.

The top section of the panel contains five input fields that define the index to be created. If you entered object name information on the Db2® object functions panel, these values are carried forward and the appropriate name fields are pre-filled.

The bottom part of the panel contains a list of options that need to be completed to define the index. There is a single selection field, in which you can type the number of the required option.

The first option is required (define the columns in the index). Typically, the third option is needed (define the space requirements for the index).

When the index has been completely specified, you select option 7 to create the index.

Related tasks

Index type (Create Index) on page 237
Index Column Selection (Create Index) on page 237

Allocation (Create Index) on page 237
Index (Create Index) on page 238
Index Partitions (Create Index) on page 238
Partition Values Option (Create Index) on page 238

Related references

Db2 Object Functions panel on page 622
Create Index panel on page 542
Create Index: Type panel on page 552
Create Index: Column Selection panel on page 546
Create Index: Allocation panel on page 545
Create Index: Options panel on page 547
Create Index: Partitions panel on page 549
Create Index: Partition Values panel on page 550

Index type (Create Index)

You can use the first group of input fields to enter uniqueness information, and the type of index.

Related references

Create Index: Type panel on page 552

Index Column Selection (Create Index)

Use this panel to select the columns that comprise the index.

Note:

Each time a column is selected using *S*, it is assigned an order one greater than the maximum order used so far.

If the column currently containing the highest order number is deselected, the maximum order is decremented (otherwise unchanged).

Because of the previous point and the ability of the user to enter an arbitrary number, gaps can occur in the order. However, any such gaps are ignored when the SQL is built.

Related references

Create Index: Column Selection panel on page 546

Allocation (Create Index)

Use this panel to specify the space requirements for the index being created.

Related references

Create Index: Allocation panel on page 545

Index (Create Index)

Use this panel to specify various YES/NO and data sharing options. In most cases, the default values are appropriate.

Related references

Create Index: Options panel on page 547

Index Partitions (Create Index)

Use this panel to specify space information for index partitions.

Related references

Create Index: Partitions panel on page 549

Partition Values Option (Create Index)

Use this panel to specify limit values for index entries in each partition.

Related references

Create Index: Partition Values panel on page 550

Creating a synonym

To create a new synonym, perform the following on the Db2® Object Functions panel:

1. Type `7` in the **Object Type** entry field.
2. Type `C` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Synonym panel.

A synonym is an alternative name for a table or view. It is used as an unqualified identifier and is specific to the SQLID that owns the synonym.

For more details, see the *Db2 UDB for z/OS SQL Reference*.

Related references

Db2 Object Functions panel on page 622
Create Synonym panel on page 560

Creating a distinct type

To create a new distinct type, perform the following on the Db2® Object Functions panel:

1. Type `8` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Distinct Type panel.

Related references

Db2 Object Functions panel on page 622
Create Distinct Type panel on page 531

Creating a function

To create a new function, perform the following on the Db2® Object Functions panel:

1. Type `9` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Function panel.

Use this panel to create an external scalar function.

Related tasks

Parameters (Create Function) on page 239
Parameter Types (Create Function) on page 240
Returned Data Type (Create Function) on page 240
Option List 1/2 (Create Function) on page 240
Option List 2/2 (Create Function) on page 240

Related references

Db2 Object Functions panel on page 622
Create Function panel on page 536
Create Function: Parameters panel on page 539
Create Function: Parameter Types panel on page 537
Create Function: Returned Data Type panel on page 540
Create Function: Option List (1/2) panel on page 533
Create Function: Option List (2/2) panel on page 534

Parameters (Create Function)

Use this panel to define and optionally name the parameters for the function.

Related references

Create Function: Parameters panel on page 539

Parameter Types (Create Function)

Use this panel to specify additional type information for function parameters.

Related references

Create Function: Parameter Types panel on page 537

Returned Data Type (Create Function)

Use this panel to specify the data type returned by the function.

Related references

Create Function: Returned Data Type panel on page 540

Option List 1/2 (Create Function)

Use this panel to specify various options to further define the function.

Related references

Create Function: Option List (1/2) panel on page 533

Option List 2/2 (Create Function)

This panel lets you specify additional options.

Related references

Create Function: Option List (2/2) panel on page 534

Creating a procedure

To create a new procedure, perform the following on the Db2® Object Functions panel:

1. Type `10` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Procedure panel.

Related tasks

Procedure Parameters (Create Procedure) on page 241
Parameter Types (Create Procedure) on page 241
Option List 1/2 (Create Procedure) on page 241
Option List 2/2 (Create Procedure) on page 241

Related references

- Db2 Object Functions panel on page 622
- Create Procedure panel on page 553
- Create Procedure: Parameters panel on page 558
- Create Procedure: Parameter Types panel on page 557
- Create Procedure: Option List (1/2) panel on page 554
- Create Procedure: Option List (2/2) panel on page 556

Procedure Parameters (Create Procedure)

This panel is equivalent to the Create Function: Parameters panel.

Related references

- Create Procedure: Parameters panel on page 558
- Create Function: Parameters panel on page 539

Parameter Types (Create Procedure)

This panel is equivalent to the Create Function: Parameter Types panel.

Related references

- Create Procedure: Parameter Types panel on page 557
- Create Function: Parameter Types panel on page 537

Option List 1/2 (Create Procedure)

This panel lets you specify various options to further define the procedure.

Related references

- Create Procedure: Option List (1/2) panel on page 554

Option List 2/2 (Create Procedure)

This panel lets you specify additional options.

Related references

- Create Procedure: Option List (2/2) panel on page 556

Creating a trigger

To create a new trigger, perform the following on the Db2® Object Functions panel:

1. Type `11` in the **Object Type** entry field.
2. Type `c` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Trigger panel.

Related tasks

Details (Create Trigger) on page 242
Search Conditions (Create Trigger) on page 242
SQL Statement (Create Trigger) on page 242

Related references

Db2 Object Functions panel on page 622
Create Trigger panel on page 596
Create Trigger: Details panel on page 598
Create Trigger: Search Condition panel on page 600
Create Trigger: SQL statement panel on page 600

Details (Create Trigger)

This panel lets you specify additional details required for the definition of a trigger.

Related references

Create Trigger: Details panel on page 598

Search Conditions (Create Trigger)

Use this panel to specify a search condition. The triggered SQL statements are run only if the search condition evaluates to true or is omitted.

Related references

Create Trigger: Search Condition panel on page 600

SQL Statement (Create Trigger)

Use this panel to specify the SQL to be run if the trigger is activated.

Related references

Create Trigger: SQL statement panel on page 600

Creating an auxiliary table

To create a new auxiliary table, perform the following on the Db2® Object Functions panel:

1. Type `12` in the **Object Type** entry field.
2. Type `C` on the command line.
3. Press Enter. ZDT/Db2 displays the Create Auxiliary Table panel.

Auxiliary tables are required for LOB support. An auxiliary table is needed if a table contains a LOB column. If such a column is defined, an auxiliary table needs to be defined to store the data for that column. If the table is also partitioned, there needs to be an auxiliary table for each partition.

Related references

Db2 Object Functions panel on page 622
Create Auxiliary Table panel on page 528

Dropping Db2® objects

The available drop functions are shown in Table 9: Supported actions for each Db2 object on page 208.

To drop a Db2® object, perform the following using the Db2® Object Functions panel:

1. Specify the relevant number (1–11) for the Db2® object type you want to drop in the **Object Type** entry field.
2. If you want ZDT/Db2 to display a Drop Confirmation panel before it actually drops the object you have specified, check the **Confirm object drop** option.
3. Type `D` on the command line.
4. Press Enter. ZDT/Db2 displays the relevant Drop panel.
5. Specify the name of the Db2® object you want to drop and press Enter.

   If you have selected the **Confirm object drop** option, ZDT/Db2 displays a Drop Confirmation panel. To proceed with the drop, press Enter. To cancel the drop, press the Cancel function key (F12) or the Exit function key (F3).

   If you have not selected the **Confirm object drop** option, ZDT/Db2 drops the specified object.

If the drop operation fails, ZDT/Db2 displays an error message; press the Help function key (F1) for further information.

Related references

Db2 Object Functions panel on page 622
Drop panels on page 640
Drop Confirmation panel on page 640

Chapter 7. Populating a Db2® table with data

You can populate a table with data by using the ZDT/Db2 Create utility (3.8). You can specify the number of rows that are to be created and how each row is to be initialized. By editing a template, you can use fill characters and patterns to initialize the data and change the data create attributes for individual columns.

When fill characters and/or patterns are used to initialize a column, the resultant data must conform to any data type restrictions imposed by Db2® for that column.

During the data creation process, ZDT/Db2 attempts to insert each row formatted by the data create utility into the target Db2® object. The insert operation may be rejected by Db2® if uniqueness, referential integrity, or other constraints would be violated.

The data create operation can be performed in background (batch execution) or in the foreground.

The main panel for the ZDT/Db2 Create utility is the Data Create Utility panel.

To display the Data Create Utility panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `8` (Create)

Now that you have the Data Create Utility panel displayed, you can specify the name of the table you want to populate with data, the number of rows that are to be created, and how each row is to be initialized:

1. Identify the Db2® table you want to populate with data using the **Specify the Db2® Object** entry fields.
2. Use the **Create Count** entry field to specify the number of rows you want to create.
3. If you are using a specific template to specify the data initialization details, (see option 1 in **Template usage**), specify the name of the template data set.
4. Use the **Processing Options** to specify which template you want to use for the create process, and whether you want to edit the template before the create process.
5. Press Enter. If you selected **Edit template**, the Column Selection/Edit panel is displayed. Otherwise, ZDT/Db2 creates the data in the specified Db2® table.

Related tasks

Specifying a Db2 object name on page 46
Specifying a data set and a member name on page 52
Editing the template on page 245

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Data Create Utility panel on page 603

Editing the template

If you selected the **Edit template** option on the Create panel, ZDT/Db2 displays the Column Selection/Edit panel when you press Enter.

To populate the columns for the Db2® table you have specified by editing the template:

1. Enter `E` against the first column in the Column Selection/Edit panel. ZDT/Db2 displays one of five Column Attributes panels depending on the data type of the column:
   - Alphanumeric (for CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC data types)
   - Numeric (for DECIMAL or NUMERIC, SMALLINT, INTEGER, REAL, and DOUBLE data types)
   - DATE
   - TIME
   - TIMESTAMP
2. Use the Column Attributes panel to specify the *create attributes* for the column. The create attributes determine the value that ZDT/Db2 puts in the column when it creates data.

   For details about how to enter the create attributes, refer to the description for the relevant Column Attributes panel.
3. Press the Exit function key (F3) to return to the Column Selection/Edit panel.
4. Repeat from Step 1 for each of the remaining columns you want to populate.

Related references

Db2 data types on page 440
Column Selection/Edit panel on page 497
Column Attributes panel (alphanumeric) on page 484
Column Attributes panel (numeric) on page 490
Column Attributes panel (DATE) on page 489
Column Attributes panel (TIME) on page 494
Column Attributes panel (TIMESTAMP) on page 495

Chapter 8. Copying data

ZDT/Db2 provides the following utilities that allow you to copy data from one place to another:

Copy Utility (3.3)

Copies data from one Db2® object to another Db2® object (that is accessible from within the same Db2® subsystem).

Import Utility (3.6)

Copies data from a VSAM or QSAM file into a Db2® table or view.

Export Utility (3.7)

Copies data from a Db2® table or view to a VSAM or QSAM file.

Related tasks

Copying data from one Db2 object to another on page 246
Copying data from a VSAM or QSAM file on page 251
Copying data to a VSAM or QSAM file on page 279
How do I...? on page 292

Copying data from one Db2® object to another

To copy data between Db2® tables, use the Copy utility function (3.3).

Note:

1. To copy data *to* a Db2® table or view from a QSAM or VSAM data set, use the Import utility function (3.6).
2. To copy data *from* a Db2® table or view to a QSAM or VSAM data set, use the Export utility function 3.7).

When you copy data from one Db2® object to another, you can:

- Select which rows to copy
- Reformat data during the copy by mapping columns in the input table to different columns in the output table
- Initialize new columns using user-specified values or patterns
- Limit the number of rows copied

The main panel for the Copy utility is the Copy Utility ("From") panel.

To display the Copy Utility ("From") panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `3` (Copy)

You can now specify details about:

- The source table (where the data is that you want to copy). During this step, you can also edit the source template to restrict the data that is copied (if applicable).
- The target table (where you want the data copied). During this step, you can also change the copy options, edit the template mapping for the target table, and specify if you want the copy performed in a batch job.

Related tasks

Copying data from a VSAM or QSAM file on page 251
Copying data to a VSAM or QSAM file on page 279
Specifying details for the "From" table on page 247
Specifying details for the "To" table on page 249

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Copy Utility ("From") panel on page 513

Specifying details for the "From" table

Specify the name of the source (*"From"*) table on the Copy Utility *"From"* panel. If the table you specify does not exist, ZDT/Db2 issues an error message and stops the utility.

If you do not want to copy all of the rows in the *"From"* table, specify the number of rows you want to copy (1–99999999) in **Copy Count**.

If you are using a specific template for the *"From"* table (see options 1 and 4 in **Template usage**), specify the name of the template data set and, optionally, the member name, in the **From Template** entry fields.

The **Processing Options** specify which template you want to use for the *"From"* table, and whether you want to edit the template before the copy process.

Template usage

The Copy Utility panel provides four options for template usage:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **From Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table. (For a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. Generate from table

Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template From** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **From Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a "/" or an "A":

Edit template

Use to edit the template for the *"From"* table before proceeding to the Copy Utility *"To"* panel.

Copy "From" values to "To" panel

Use to copy the values (Db2® object **Location**, **Owner**, **Name**, and template **Data set name**, **Member**) entered on the *"From"* panel to the corresponding fields on the *"To"* panel.

To proceed, press Enter.

If you selected **Edit template** (either by using a "/", or by having previously used the *"A"* selection character for this option), the Column Selection/Edit panel is displayed. Otherwise, the Copy Utility *"To"* panel is displayed.

Related tasks

Specifying a Db2 object name on page 46
Selecting options on ZDT/Db2 panels on page 45
Working with templates on page 74
Editing the template for the "From" table on page 248
Specifying details for the "To" table on page 249

Related references

Copy Utility ("From") panel on page 513
Column Selection/Edit panel on page 497
Copy Utility ("To") panel on page 521

Editing the template for the "From" table

If you select the **Edit template** option on the Copy Utility *"From"* panel, ZDT/Db2 displays the Column Selection/Edit panel (Figure 59: Primary Column Selection/Edit panel on page 249). You can use this panel to restrict the data in the *"From"* table that you want copied by:

- Selecting (or deselecting) specific columns for copying.
- Specifying *row selection criteria*. Only rows whose contents satisfy the criteria are selected for copying.

For example, the template shown in Figure 59: Primary Column Selection/Edit panel on page 249 has been edited to select certain columns and only rows whose contents meet certain criteria. If you use this template as the *"From"* template for your Copy, the data copied is restricted to columns where WORKDEPT contains *"MNT"* and JOB contains *"PAINTER"*.

Figure 59. Primary Column Selection/Edit panel

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)              Column Selection/Edit                Line 1 of 16
TABLE HFMUSER.EMP
------- Row Selection Criteria ---- (Use SQL/PF4 for full screen edit) -------
1 Sel: WHERE WORKDEPT = 'MNT' AND JOB = 'PAINTER'                            +

Cmd Seq SHEX CL# Column name          Data type(length)  Null Default  Order A/D
                 ****  Top of data  ****
___ ___ S      1 EMPNO                CHARACTER(6)            None       ___   _
___ ___ S      2 FIRSTNME             VARCHAR(12)             None       ___   _
___ ___ S      3 MIDINIT              CHARACTER(1)            None       ___   _
___ ___ S      4 LASTNAME             VARCHAR(15)             None       ___   _
___ ___ S      5 WORKDEPT             CHARACTER(3)        Y   Null       ___   _
___ ___ S      6 PHONENO              CHARACTER(4)        Y   Null       ___   _
___ ___ S      7 HIREDATE             DATE                Y   Null       ___   _
___ ___ S      8 JOB                  CHARACTER(8)        Y   Null       ___   _
___ ___ S      9 EDLEVEL              SMALLINT            Y   Null       ___   _
___ ___ S     10 SEX                  CHARACTER(1)        Y   Null       ___   _
___ ___ S     11 BIRTHDATE            DATE                Y   Null       ___   _
___ ___ S     12 SALARY               DECIMAL(9,2)        Y   Null       ___   _
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=SQL       F5=RFind     F6=RunTemp
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Working with templates on page 74

Related references

Copy Utility ("From") panel on page 513
Column Selection/Edit panel on page 497
Copy Utility ("To") panel on page 521

Specifying details for the "To" table

Specify the name of the target (*"To"*) table on the Copy Utility *"To"* panel.

If you are using a specific template for the *"To"* table, specify the name of the template data set and, optionally, the member name, in the **To Template** entry fields.

The **Processing Options** specify which template you want to use for the *"To"* table, and whether you want to edit the template before the copy process.

Template usage

The Copy Utility panel provides four options for template usage:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **To Template** section of the panel. If you do not specify a member name, or if you

specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table. (For a detailed explanation, see Template Usage option 3).

2. Previous

Uses the last (previously used) template for this table.

3. Generate from table

Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **To Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **To Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a **"/"** or an "A":

Edit copy options

Use to edit the Copy options before copying the data.

Edit template mapping

Use to edit the template mapping for the *"To"* table before copying the data.

Batch execution

Builds the necessary JCL to perform the Copy utility in a batch job.

To proceed, press Enter.

If you selected **Edit Copy options** (either by using a **"/"**, or by having previously used the *"A"* selection character for this option), the Copy Options panel is displayed for editing.

If you selected **Edit template mapping** (either by using a **"/"**, or by having previously used the *"A"* selection character for this option), the Template Mapping panel is displayed for editing.

If you selected **Batch execution**

(either by using a "/", or by having previously used the *"A"* selection character for this option), the generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

Otherwise (if you selected none of the preceding options), ZDT/Db2 copies the data from the *"From"* table to the *"To"* table.

Setting options for the current Copy session

If you select **Edit copy options** on the Copy *"To"* panel ZDT/Db2 displays the Copy Options panel.

You can use this panel to change the options for the current Copy session.

Note: Any options you set on this panel apply for the current Copy session only. When you exit from the Copy session, the copy options revert to the global copy options.

Related references

Copy Options panel on page 509

Running Copy in batch

If you want to copy large Db2® tables, you can perform the copy in a batch job.

If you select the **Batch execution** option on the Copy Utility *"From"* panel, the Copy utility builds the necessary JCL to perform the copy function in a batch job. The generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

Related tasks

Specifying a Db2 object name on page 46
Selecting options on ZDT/Db2 panels on page 45
Setting options for the current Copy session on page 251
Mapping data on page 112
Running Copy in batch on page 251

Related references

Copy Utility ("To") panel on page 521
Copy Utility ("From") panel on page 513
Copy Options panel on page 509
Copy utility options (option 3.3) on page 68

Copying data from a VSAM or QSAM file

To copy data to a Db2® table or view from a QSAM or VSAM data set, use the Import utility function (3.6).

Note:

1. To copy data between Db2® tables, use the Copy utility function (3.3).
2. To copy data *from* a Db2® table or view to a QSAM or VSAM data set, use the Export utility function 3.7).

The input to the Import utility function can be a QSAM or VSAM data set produced by:

- The ZDT/Db2 Export utility
- Any external source

You must specify a copybook or template that describes the record format of the data in the input QSAM or VSAM data set. If you specify a copybook, ZDT/Db2 compiles it to produce a template.

When you copy data to a Db2® table or view from a QSAM or VSAM data set, you can:

- Reformat data during import by mapping Db2® table columns to fields in the input QSAM or VSAM data set (as defined by a copybook or template).
- Use a REXX procedure to enhance the way in which data is selected and imported.
- Specify the number of rows to be imported, and the starting record in the import file.
- Specify whether you want to run the import in foreground or batch.
- Specify that any duplicate rows are to be updated with the values in the imported row.
- Specify how often ZDT/Db2 should commit changes to Db2® during the import process.

The main panel for the Import utility is the Import Utility ("From") panel.

To display the Copy Utility ("From") panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `6` (Import)

You can now specify details about:

- The source QSAM or VSAM data set (where the data is that you want to copy). During this step, you can also edit the source template to restrict the data that is copied (if applicable).
- The target table (where you want the data copied). During this step, you can also:
    - Edit the template mapping for the target table
    - Specify if you want the copy performed in a batch job
    - Optionally specify the name of a REXX procedure to further enhance the way data is selected

When (and when not) to use the Import utility

The Import utility provides one method for moving Db2® data between different Db2® subsystems (that is, export on one subsystem followed by import on another). If you want to move data between tables in the same Db2® sub-system, either the Copy utility or direct execution of SQL statements is typically more convenient.

In general terms, the Import utility is suitable for inserting small to medium volumes of data into Db2® tables. Import operations are achieved by inserting each selected record in the import data set into the target table, and any update operations are achieved by generating an update statement.

All insert and update operations are logged by Db2®, which can have operational and performance implications. The Import utility is not intended as a substitute for the Db2® utilities if unloading or migrating large volumes of data, and the Db2® utilities offer superior performance in these situations. Whenever possible, run import operations in batch.

The import utility provides for automatic mapping of the Db2® null indicator, and, optionally, one of the following formats can be specified corresponding to the template used:

ZDT/Db2 (SQLDA) format
Db2® unload format
DSNTIAUL format
User defined format (excluding Placement, User defined)

However, the template editor does not show the null indicator as a separate item in the "From" template on the Field Mapping panel. It is therefore not possible to re-map the null indicator associated with nullable columns to a column in the target table.

Related tasks

Compiler language selection (option 0.4) on page 65
Specifying details for the "From" data set on page 253
Specifying details for the "To" table on page 255
Using a REXX procedure with the Import utility on page 257

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Import Utility ("From") panel on page 719
Import Utility ("To") panel on page 720

Specifying details for the "From" data set

Specify the name of the source (*"From"*) data set on the Import Utility *"From"* panel. Use a combination of the **Data set name**, **Member**, and **Volume** entry fields to specify the sequential data set, VSAM data set, or PDS member from which you want to import the data.

If you do not want to import all of the records from the input data set, specify the number of records you want to import (1–99999999) in **Import Count**.

If you are using a specific template for the *"From"* data set (see option 1 in **Template usage**), specify the name of the template data set and, optionally, the member name, in the **From Copybook or Template** entry fields.

The **Processing Options** on the Import Utility "From" panel specify template usage, and whether you want to edit the template before the import process.

Template usage

The Import Utility *"From"* panel provides two options for template processing:

1. **Above**

   Requires that you enter the name of a copybook or template in the **From Copybook or Template** section of the panel. After you have specified a name of a copybook or template, processing continues using this template in place of the previously used template (see Template usage option 2).

2. **Previous**

   Uses the last (previously used) template for this data set.

You can select the following option by entering a "/" or an "A":

Edit template

Use to edit the template for the *"From"* data set before proceeding to the Import Utility *"To"* panel.

To proceed, press Enter.

If you selected **Edit template** (either by using a "/", or by having previously used the *"A"* selection character for this option), the Field Selection/Edit panel is displayed.

Otherwise, the Import Utility *"To"* panel is displayed.

Related tasks

Specifying a data set and a member name on page 52
Selecting options on ZDT/Db2 panels on page 45
Working with templates on page 74
Editing the template for the "From" data set on page 254
Specifying details for the "To" table on page 255

Related references

Import Utility ("From") panel on page 719
Import Utility ("To") panel on page 720

Editing the template for the "From" data set

If you select the **Edit template** option on the Import Utility *"From"* panel, ZDT/Db2 displays the Field Selection/Edit panel (see Figure 60: Import Utility: Field Selection/Edit panel for the "From" data set on page 255).

You can narrow down which records you want to process by specifying *record selection criteria*. Only records whose contents satisfy the criteria are selected. You specify record selection criteria using REXX comparison expressions.

Note: Record selection criteria is applied on the input data *before* any field mapping or data conversion is performed, whereas the REXX import procedure applies *after* any field mapping or data conversion is performed.

For example, the template shown in Figure 60: Import Utility: Field Selection/Edit panel for the “From” data set on page 255 has been edited to select only records that meet certain criteria. If you use this template as the *“From”* template for your Import utility, the data imported is restricted to records where SALARY contains a value greater than 50 000.

Figure 60. Import Utility: Field Selection/Edit panel for the “From” data set

```
 Process   Options   Help
------------------------------------------------------------------------------
Z Data Tools                 Field Selection/Edit                 Line 1 of 10

----------- Criteria - Enter 1 or 2 to specify expression by field -----------
1 Id :                                                                        +
2 Sel: #6 > 50000                                                             +
Offset     5
Cmd Seq SHE Ref Field Name                              Picture  Type Start Length
              ****  Top of data  ****
___ ___ S     1 1 REC-TYPE01                                      AN      1     85
___ ___ S     2  2 REC-TYPE                             XX        AN      6      2
___ ___ S     3  2 NAME                                 X(20)     AN      8     20
___ ___ S     4  2 EMPLOYEE-NO                          9(4)      BI     28      2
___ ___ S     5  2 AGE                                  9(4)      BI     30      2
___ ___ S     6  2 SALARY                               9(7)      PD     32      4
___ ___ S     7  2 MONTH OCCURS 12 TIMES                9(8)      BI     36      4
___ ___ S     8  2 FILLER                               XX        AN     84      2
              ****  End of data  ****


Command ===> ____________________________________________________ Scroll HALF
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

If you press Enter, ZDT/Db2 saves the updated template and returns you to the Import Utility “From” panel.

Note: For more information about specifying record selection criteria, see the Z Data Tools User’s Guide and Reference.

Related tasks

General information about REXX on page 258

Related references

Import Utility ("From") panel on page 719
Field Selection/Edit panel on page 704

Specifying details for the "To" table

Specify the name of the table to which the input data is to be imported on the Import Utility *“To”* panel. If the table you specify does not exist, ZDT/Db2 issues an error message.

If you are using a specific template for the *“To”* table (see options 1 and 4 in **Template usage**), specify the name of the template in the **To Template** entry fields.

To delete all the rows in the target table before importing the data from the QSAM or VSAM data set, select the **Delete existing rows** option.

Set the **Duplicate row options** according to how you want ZDT/Db2 to handle any instances where the data to be imported is a *"duplicate"* of an existing row. That is, where the only changes to the data are in columns that are not part of any unique index key:

- To ignore duplicate rows, select **1. Ignore**.

  Any duplicates that ZDT/Db2 detects are not updated and ZDT/Db2 ignores the value (if any) in the **For … duplicates** field.
- To update duplicate rows, select **2. Update**.

  ZDT/Db2 updates any duplicate rows. If the number of duplicates exceeds the value specified in the **For … duplicates** field, ZDT/Db2 terminates the import operation and all changes are rolled back.

To proceed, press Enter.

If you selected **Edit template mapping** (either by using a "/", or by having previously used the *"A"* selection character for this option), then, before ZDT/Db2 imports the data, the Template Mapping panel is displayed for editing.

If you selected **Batch execution** (either by using a "/", or by having previously used the *"A"* selection character for this option), the generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

Otherwise, ZDT/Db2 imports the data from the *"From"* data set to the *"To"* Db2® table.

Related tasks

Specifying a Db2 object name on page 46
Specifying a data set and a member name on page 52
Mapping data on page 112
Running Import in batch on page 256

Related references

Import Utility ("To") panel on page 720
Template Mapping panel on page 900

Running Import in batch

If you want to import large Db2® tables, you can perform the import in a batch job.

If you select the **Batch execution** option on the Import Utility *"To"* panel, the Import utility builds the necessary JCL to perform the import function in a batch job. The generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

If you have one or more data sets that contain the template that describes the record structure of your input data, you can specify the data set details (or TSO ALLOC statement) in the generated JCL with the keywords TINPUT or TCIN.

Similarly, if you have one or more data sets that contain the template that describes the record structure of your output data, you can specify the data set details (or TSO ALLOC statement) in the generated JCL with the keywords TOUTPUT or TCOUT.

Related references

DBI (Import) batch command on page 1034
Import Utility ("To") panel on page 720

Using a REXX procedure with the Import utility

You can specify an existing REXX procedure, or create a new one, to further enhance the way in which data is selected and imported.

You can code any number of REXX statements and functions to manipulate the output record, select specific records, print reports and tally numeric values. If you are coding a REXX procedure, be aware that:

- The import procedure is run after input template record selection is performed. The import procedure can cause a record (which would have been selected by the template selection process, if there was one) to be deselected for copying using RETURN DROP.
- The import procedure is run after any field mapping or conversion supplied by a template has been performed.
- If the record has been reformatted as a result of the template processing, the variable INREC contains the input record value and the variable OUTREC contains the reformatted output record.
- Changes made by the import procedure to INREC are ignored. Changes made to OUTREC are used in the import process to load the Db2® table.
- If using the PRINT() function, the template is determined as follows:
  - If an input template without an output template has been specified, the input template is used.
  - If an output template is specified, and the record value specified to print is the input record, the input template is used. Otherwise the output template is used.
  - If a record cannot be matched against the selected printing template, it is not printed.

Example 1

Import records, inserting a new field after column 40 for a length of 4.

```
outrec = fld(1,40)||'    '||fld(41)
```

Example 2

Import records with type A and print the first 10. The type value is in column 6.

```
if fld(6,1) = 'A' then do                       /* Type A?  */
   If prtcount() <= 10 then print(inrec,'CHAR') /* Print 10 */
   RETURN                                       /* Copy     */
end
return 'DROP'                                   /* Drop the rest */
```

Example 3

Change the output records to uppercase while importing.

```
upper outrec
```

Related tasks

Specifying details for the "To" table on page 255
Enhancing the Import utility using a REXX procedure on page 268

General information about REXX

REXX stands for the REstructured eXtended eXecutor language. REXX is a general purpose programming language, similar to PL/I. It includes extensive parsing capabilities for character manipulation, extensive mathematical capabilities, and numerous built-in functions that perform various processing, searching, and comparison functions.

REXX programs are typically run by a language processor (interpreter). That is, the program is run statement-by-statement, without first being converted to another form (compiled). Because of the interpretive nature of REXX, ZDT/Db2 is able to exploit the power of REXX when processing record selection criteria.

Although REXX is a powerful and versatile language, coding selection criteria using REXX is straightforward. You do not need to know how to write REXX programs. All you need to know is how to write REXX comparison expressions. ZDT/Db2 takes care of turning your selection criteria into a REXX program.

Much of the information in this section has been extracted from the *OS/390® TSO/E REXX Reference* and *OS/390® TSO/E REXX User's Guide*. For more detailed information about REXX, see those manuals.

Overview of REXX expressions

REXX expressions consist of one or more *terms* interspersed with zero or more *operators* that denote operations to be carried out on terms. Expressions are evaluated left to right, modified by parentheses and by operator precedence in the typical algebraic manner. If parentheses are encountered (other than those that identify function calls) the entire subexpression between the parentheses is evaluated immediately when the term is required. Expressions are wholly evaluated, unless an error occurs during evaluation. The REXX language uses a free format. This means you can insert extra spaces between words without causing an error.

The following list shows the terms you can use in a selection criteria expression. The operators you can use are described on the pages that follow.

Literal strings

A literal string is a sequence including any characters and delimited by the single quotation mark (`'`) or the double quotation mark (`"`). Use two consecutive double quotation marks (`""`) to represent a `"` character within a string delimited by double quotation marks. Use two consecutive single quotation marks (`''`) to represent a `'` character within a string delimited by single quotation marks. A literal string is a constant and its contents are never modified when it is processed.

These are valid strings:

```
'Fred'
"Don't Panic!"
'You shouldn''t'        /* Same as "You shouldn't" */
```

Note that a string followed immediately by a `(` is considered to be the name of a function. If followed immediately by the symbol `X` or `x` it is considered to be a hexadecimal string. If followed immediately by the symbol `B` or `b` it is considered to be a binary string.

A hexadecimal string is a literal string, expressed using a hexadecimal notation of its encoding. It is any sequence of zero or more hexadecimal digits (0-9, a-f, A-F), grouped in pairs. A single leading 0 is assumed, if necessary, at the front of the string to make an even number of hexadecimal digits. The groups of digits are optionally separated by one or more blanks, and the whole sequence is delimited by single or double quotation marks, and immediately followed by the symbol X or x. (Neither x nor X can be part of a longer symbol. A hexadecimal string is a literal string formed by packing the hexadecimal digits given. Packing the hexadecimal digits removes blanks and converts each pair of hexadecimal digits into its equivalent character, for example: `'C1'X A.`

You can use hexadecimal strings to include characters in a program even if you cannot directly enter the characters themselves. These are valid hexadecimal strings:

```
'ABCD'x
"1d ec f8"X
"1 d8"x
```

A binary string is a literal string, expressed using a binary representation of its encoding. It is any sequence of zero or more binary digits (0 or 1) in groups of 8 (bytes) or 4 (nibbles). The first group can have fewer than four digits; in this case, up to three 0 digits are assumed to the left of the first digit, making a total of four digits. The groups of digits are optionally separated by one or more blanks, and the whole sequence is delimited by matching single or double quotation marks and immediately followed by the symbol b or B.

A binary string is a literal string formed by packing the binary digits given. If the number of binary digits is not a multiple of eight, leading zeros are added on the left to make a multiple of eight before packing. You can use binary strings to specify characters explicitly, bit by bit.

These are valid binary strings:

```
'11110000'b        /* == 'f0'x                  */
"101 1101"b        /* == '5d'x                  */
'1'b               /* == '00000001'b and '01'x  */
'10000 10101010'b  /* == '0001 0000 1010 1010'b */
''b                /* == ''                     */
```

Symbols

Character strings, without quotation marks, which are converted to uppercase. Any symbol that begins with a `#` is treated as a reference to a field in the record being processed, and the value of the field is used. All other symbols are treated as constants.

ZDT/Db2 assigns to each field defined in a template a unique field reference number. If you want to refer to a field in a selection criteria expression, you specify the field's field reference number prefixed by `#.` Note that you can only refer to fields defined in the record you are currently working with.

REXX expression evaluation only processes data in the form of *"typeless"* character strings (typeless because they are not, as in the case of COBOL, of a particular data type such as binary, packed-decimal, and so forth). Therefore, if you refer to a numeric field, ZDT/Db2 converts the value of the field to a numeric character string that can be processed by REXX when evaluating the selection criteria expression. The number of integer and decimal digits in the character string is determined by the field definition in your template. For example, if you refer to a packed-decimal field with a COBOL PICTURE clause of 999V99, ZDT/Db2 converts the value of the field to a character string consisting of numeric digits, a period for the virtual decimal place, and, if the value of the field is negative, a leading sign character (such as -123.45). Note that all numeric fields are treated as signed, regardless of whether the COBOL PICTURE clause contains a sign symbol.

Occasionally, you might want to evaluate the value of a numeric field without converting it to a numeric character string. To do this, prefix the field reference number by `#u` instead of `#.` This tells ZDT/Db2 not to convert the number to a numeric character string. For example, if to test a 2-byte binary numeric field (with a field reference number of 45) for a special value of X'FFFF', code:

```
#u45 = 'FFFF'x
```

If you refer to a field in an array, you must qualify the field reference with the required number of subscripts enclosed in parentheses and separated by commas. The number of subscripts you specify must equal the number of dimensions in the array that contains the field you are referencing. In COBOL terms, there must be a subscript for each OCCURS clause in the hierarchy containing the field, including any OCCURS clause for the field itself. Each subscript you specify must be a positive integer in the range 1 to the maximum number of occurrences of the field as specified in the OCCURS clauses. If the field you refer to is in a variable-length array (specified using the OCCURS DEPENDING ON clause), ensure that you do not refer to an occurrence of the field that, for any given record, might not exist. If you do refer to a non-existent field, the selection criteria is not satisfied, and the record is not selected.



Note: The object of an OCCURS DEPENDING ON clause must be defined as a field in the static portion of the same record as the array (that is, the portion of the record that precedes any variable-length array defined with the OCCURS DEPENDING ON clause). If the object of the OCCURS DEPENDING ON clause is not defined in this way, you cannot refer to any fields in the array or any fields in the record that follow the array.

Function call

A call to REXX built-in function.

Subexpressions

Terms in an expression bracketed within left and right parentheses.

Comparison operators

REXX comparison operators compare two terms and return the value 1 if the result of the comparison is true, or 0 otherwise. Comparison operators can compare numbers or character strings. The most commonly used comparison operators are:

Operator	Meaning
=	Equal
¬=, \=	Not equal
<	Less than
¬<, \<	Not less than
>	Greater than
¬>, \>	Not greater than
<=	Less than or equal to
>=	Greater than or equal to
><	Greater than or less than (same as not equal)
<>	Less than or greater than (same as not equal)

Note that the `not` character (¬), is synonymous with the backslash (\). You can use the two characters interchangeably.

When comparing terms using these comparison operators, if both terms in the expression are numeric, REXX performs a numeric comparison. Otherwise, both terms are treated as character strings and REXX performs character comparison. (A number in REXX is a string that contains one or more decimal digits, an optional decimal point, and an optional leading sign character. The string can contain one or more leading or trailing blanks, and the sign character can be separated from the digits by one or more blanks.)

In a character comparison, leading and trailing blanks are ignored, and the shorter string is padded with blanks on the right. Character comparisons are case-sensitive. To prevent lowercase characters being converted to uppercase, delimit character

strings with quotation marks. For example, if the field `#4` contains the value `MixedCase` both the following comparison operations would be true:

```
#4 = 'MixedCase'
#4 = '    MixedCase     '
```

but the following comparison operation would not be true:

```
#4 = MixedCase
```

In numeric comparisons, the comparison is effected by subtracting the two numbers (calculating the difference) and then comparing the result with 0. For example, the comparison operation:

```
#6 = 10
```

is identical to the operation:

```
(#6 - 10) = 0
```

In addition to these comparison operators, REXX provides a number of *"strict"* comparison operators that are mainly intended for use if comparing character strings. The strict comparison operators all have one of the characters defining the operator doubled, such as `==` (strictly equal).

The strict comparison operators are:

Operator	**Meaning**
==	Strictly equal
¬==, \==	Strictly not equal
<<	Strictly less than
¬<<, \<<	Strictly not less than
>>	Strictly greater than
¬>>, \>>	Strictly not greater than
<<=	Strictly less than or equal to
>>=	Strictly greater than or equal to

If you use the `==` comparison operator (strictly equal), the two character strings being compared must be identical (character by character) and of the same length to be considered strictly equal. Leading and trailing blanks are significant. For example, continuing the example using field `#4` that contains the value `MixedCase` only the first of the following comparison operations would be true:

```
#4 == 'MixedCase'
#4 == '    MixedCase     '
```

Similarly, the strict comparison operators such as `>>` or `<<` carry out a simple character-by-character comparison, with no padding of either of the strings being compared. The comparison of the two strings is from left to right. If one string is shorter than and is a leading substring of another, then it is smaller than (less than) the other. The strict comparison operators do not attempt to perform a numeric comparison on the two terms. Do not use them to compare numeric fields.

Arithmetic operators

You can process numeric terms in comparison expressions using the arithmetic operators:

Operator

Meaning

+

Add

-

Minus

*****

Multiply

/

Divide

%

Integer divide (divide and return the integer part of the result)

//

Remainder (divide and return the remainder--not modulo, because the result might be negative)

******

Power® (raise a number to a whole-number power)

Prefix -

Same as the subtraction: `0 - number`

Prefix +

Same as the addition: `0 + number`

You can use these operators to produce an intermediate result that you can compare with another term. For example, given these fields:

#6

Contains a numeric value representing an employee's annual salary.

#15

Contains a numeric value representing his annual travel allowance.

#23

Contains a numeric value representing his annual bonus.

you can use this comparison to select records for employees with a combined annual payment of greater than $100,000:

```
(#6 + #15 + #23) > 100000
```

For another example, with these fields:

#45

Contains the number of sick days an employee is entitled to annually.

#46

Contains the number of sick days an employee has used in the current year.

you can use this comparison to select records for employees who have used 50% or more of their sick days entitlement:

```
(#46 / #45) >= .5
```

Note that in each of these examples, the arithmetic subexpression is contained in parentheses. This ensures that the entire subexpression is evaluated before the comparison operation.

The order of precedence of arithmetic operators is as follows (highest is at the top):

Operator

Meaning

**+ - ¬ **

Prefix operators

******

Power®

*** /**

Multiply and divide

+ -

Add and subtract

For example, * (multiply) has a higher priority than + (add), therefore `3+2*5` evaluates as `13` (rather than the `25` that would result if strict left to right evaluation occurred). To force the addition to occur before the multiplication, rewrite the expression as `(3+2)*5.` Adding the parentheses makes the first three tokens a subexpression.

Logical (boolean) operators

REXX comparison expressions return a true (1) or false (0) value when processed. Logical operators combine two comparisons and return the true (1) or false (0) value depending on the results of the comparisons.

The logical operators are:

Operator

Meaning

&

AND

Returns 1 if both comparisons are true. For example:

```
(4 > 2) & (a = a)        /* true, therefore result is 1  */
(2 > 4) & (a = a)        /* false, therefore result is 0 */
```

|

Inclusive OR

Returns 1 if at least one comparison is true. For example:

```
(4 > 2) | (5 = 3) /* at least one is true, therefore result is 1 */
(2 > 4) | (5 = 3) /* neither one is true, therefore result is 0 */
```

&&

Exclusive OR

Returns 1 if only one comparison (but not both) is true. For example:

```
(4 > 2) && (5 = 3) /* only one is true, therefore result is 1 */
(4 > 2) && (5 = 5) /* both are true, therefore result is 0 */
(2 > 4) && (5 = 3) /* neither one is true, therefore result is 0 */
```

**Prefix ¬ **

Logical NOT

Returns the opposite response. For example:

```
\ 0              /* opposite of 0, therefore result is 1 */
\ (4 > 2)        /* opposite of true, therefore result is 0 */
```

Logical expressions are used in complex comparisons and can act as checkpoints to stop unwanted conditions (such as testing a field for a value of zero before using it as a divisor). If you have a series of logical expressions, for clarification, use one or more sets of parentheses to enclose each expression. For example:

```
(#46 = 999) | ((#45 > 0) & (#46 / #45) >= .5)
```

Useful functions

REXX provides a rich set of built-in functions, including character manipulation and conversion functions. Some of these functions might be of use when you are writing your comparison expressions. To call a function, type the function name directly followed by one or more arguments within parentheses. There can be no space between the function name and the left parenthesis. For example:

```
function(arguments)
```

A function call can contain up to 20 arguments separated by commas. Each argument can be:

Argument	Example
Blank	*function*( )
Constant	*function*(55)
Symbol	*function*(#5)
Literal string	*function*('With a literal string')
Option recognized by function	*function*(option)
Another function	*function*(*function*(*arguments*))
Combination of argument types	*function*('*Literal string*', #5, *option*)

Some of the built-in functions provided by REXX that you might find useful are:

ABS()

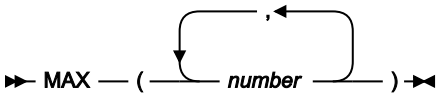
Figure 61.

Returns the absolute value of a *number.* For example, to select records in which field `#12` contains a value in the range -10 to +10, specify:

```
ABS(#12) <= 10
```

MAX()

Figure 62.

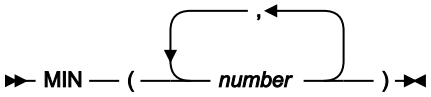


Returns the largest number from the list specified. For example, to select records in which any of fields `#10,` `#11,` or `#12` contains a value greater than 55, specify:

```
MAX(#10, #11, #12) > 55
```

MIN()

Figure 63.

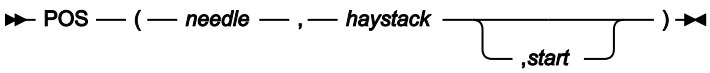


Returns the smallest number from the list specified. For example, to select records in which any of fields `#10,` `#11,` or `#12` contains a value less than 0, specify:

```
MIN(#10, #11, #12) < 0
```

POS()

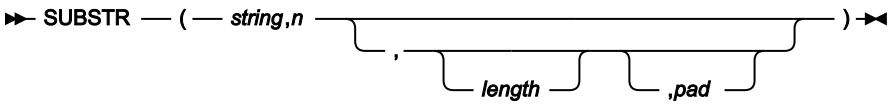
Figure 64.



Returns the position of one string, *needle,* in another, *haystack.* Returns 0 if *needle* is a null string, or is not found in *haystack,* or if *start* is greater than the length of *haystack.* By default, the search starts at the first byte of *haystack* (that is, the value of *start* is 1). You can override this by specifying *start* (which must be a positive whole number), the point at which the search starts. For example, to select records in which any character in field `#22` is a blank, specify: `POS(' ',#22) > 0`

SUBSTR()

Figure 65.



Returns the substring of *string* that begins at the *n*th byte and is of length *length* bytes, padded with *pad* if necessary. *n* is a positive whole number. If *n* is greater than the length of *string* (in bytes), only pad characters are returned.

If you omit *length,* the rest of the string is returned. The default *pad* character is a blank.

For example, to select records in which bytes 4-6 of field `#22` are the string 'NOT', specify:

```
SUBSTR(#22,4,3) = 'NOT'
```

Performance tips

Do not use EXITs in REXX

Use RETURN to terminate all your REXX processing. Do not use EXIT statements. EXIT statements in REXX used with ZDT/Db2 cause excessive CPU usage.

Examples

The following are examples of REXX comparison expressions:

Example 1

Select records in which the transaction date (field #14) is any date in July 2000, or the transaction value (field #27) is greater than $100,000.00

```
(#14 >= 20000701 & #14 < 20000801) | #27 > 100000.00
```

Example 2

Select records in which the count of credit notes (field #62) is greater than 10% of the count of invoices (field #61)

```
#62 > #61/10
```

Example 3

Select records in which either the transaction value (field #27) is greater than $50,000.00 and the purchase order number (field #25) starts with characters other than 'TX', or the transaction value is greater than $70,000 and the supplier number (field #23) is 984545, but not if both sets of conditions are true

```
(#27 > 50000.00 & SUBSTR(#25,1,2) ¬= 'TX') &&
(#27 > 70000.00 & #23 = 984545)
```

Enhancing the Import utility using a REXX procedure

When you use the Import utility, you can supply a REXX procedure to enhance typical import ZDT/Db2 processing according to your own custom requirements.

Supplying a REXX procedure when using the Import utility

You use the **Use REXX proc** field on the Import *"From"* panel to supply the REXX procedure:

```
/ Use REXX proc member
```

If you enter an * (asterisk) as the *member* name, ZDT/Db2 displays an edit panel where you can enter a temporary procedure for one time use. Use this method only for short, ad hoc procedures that you do not want to keep for future use.

Alternatively, you can allocate an HFMEXEC DD that identifies a PDS where your saved REXX procedures reside. If you specify a member name, Z Data Tools either edits a new member or runs with an existing member. Specifying blank or a

pattern other than * displays a member selection list of the PDS allocated to HFMEXEC. Concatenated HFMEXEC data sets are not supported under ISPF.

For example, if you store your REXX procedures in a PDS called 'USERID.HFMEXEC' then, before using the **Use REXX proc** field, you must issue a TSO ALLOC command similar to this:

```
TSO ALLOC DD(HFMEXEC) DSN('USERID.HFMEXEC') SHR
```

In each case, ZDT/Db2 validates the REXX procedure that you supply. If errors are detected, ZDT/Db2 shows all detected errors in a message box and opens the procedure in an edit panel which you can use to correct it. Canceling the Edit session clears the **Use REXX proc** field and redisplay the entry panel.

How enhanced processing works

The REXX procedure that you supply is run against each input record in sequence, either until the end of the input data set is reached, or until the REXX procedure issues a RETURN STOP (or RETURN STOP IMMEDIATE) command.

ZDT/Db2 defines two special REXX variables, INREC and OUTREC, that you can use in the REXX procedure that you supply to perform enhanced processing. When the ZDT/Db2 function or panel calls the REXX procedure, the contents of each input record selected for processing are passed to the procedure in both INREC and OUTREC. When the procedure is called, the contents of INREC and OUTREC are identical.

The variable INREC is intended to be used as a reference variable (ZDT/Db2 ignores any changes to INREC). The variable OUTREC can be updated by the procedure, and (unless you *"drop"* the record from further processing, as described later in this section) when the REXX procedure has completed, is passed back for processing by the Import utility. For example, the following code changes a two-digit year in the output record to a four-digit year:

```
/* Changes a date of format MMDDYY (starting at column 1) to MMDDYYYY */
If fld(5,2,z) > 50
   outrec = fld(1,4)||'19'||fld(5)
else
   outrec = fld(1,4)||'20'||fld(5)
```

Using templates with enhanced processing

Only those records that the template selects are passed to the REXX procedure for further processing.

Regardless of whether or not the template you are using selects only certain fields, the INREC and OUTREC variables contain complete records with all fields.

Related tasks

Working with templates on page 74

Dropping a record from further processing

If the REXX procedure exits normally after processing a record, that record is considered to be *"selected"*, and the contents of OUTREC are used as the output record for importing. However, if the REXX procedure ends with a RETURN DROP (or RETURN STOP IMMEDIATE) instruction, the current record is *"dropped"* from further processing by ZDT/Db2, and is not imported.

External REXX functions

ZDT/Db2 provides the following external REXX functions that you can use in addition to the built-in REXX functions described in the *OS/390® TSO/E REXX Reference*.

CHANGE

Change a character string

CONTAINS

Check for character values

FLD

Refer to a field from the current input record

NCONTAIN

Check for numeric values

PRINT

Print a record

PRTCOUNT

Count of records printed

RECSIN

Count of records read

RECSOUT

Count of records written to a given data set

TALLY

Total a field value and report the total

TM

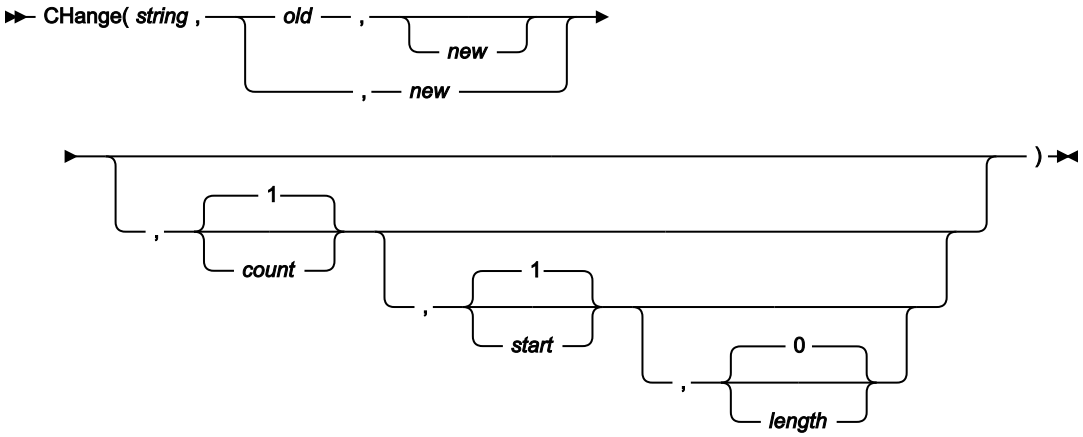
Test a string for a bit value

WRITE

Write a record

Note: You can only use these ZDT/Db2-specific REXX external functions, and the INREC and OUTREC variables, in a REXX procedure specified by the **Use REXX proc** field). You cannot use these functions and variables in REXX procedures outside of this ZDT/Db2 environment.

CHANGE



Returns *string*, with up to *count* occurrences of the substring *old* changed to the substring *new*. The default value for *count* is 1. If you specify *count*, it must be zero or a positive whole number. If you specify zero, all occurrences of *old* in *string* are changed to *new*.

You can restrict the byte positions within *string* that are examined for substring *old* by specifying a starting byte position (*start*) and a length in bytes (*length*). The default value for *start* is 1. If you specify *start*, it must be a positive whole number. If *start* is greater than LENGTH(*string*), the CHANGE function has no effect. If you specify *length*, it must be zero or a positive whole number. If you omit *length*, or specify zero, the remainder of *string* from byte position *start* is examined. If *length* is less than LENGTH(*old*), the CHANGE function has no effect.

The lengths of substrings *old* and *new* can be different. If you omit *old*, the substring *new* is inserted in *string* at the byte position indicated by *start*. If you omit *new*, the substring *old* is deleted from *string*.

Here are some examples:

Example 1

```
CHANGE('abcabcabc','abc','DeF')      ⟶    'DeFabcabc'
       /*  1 (default) occurrence of old changed          */
```

Example 2

```
CH('abcabcabc','abc','DeF',2)        ⟶    'DeFDeFabc'
       /*  2 occurrences of old changed                   */
```

Example 3

```
CHANGE('abcabcabc','abc','DeF',0)    ⟶    'DeFDeFDeF'
      /*  count = 0, all occurrences of old changed       */
```

Example 4

```
CH('abcabcabc','abc','DeF',,4)       ⟶    'abcDeFabc'
       /*  1 (default) occurrences of old changed,          */
       /*  starting at position 4                            */
```

Example 5

```
CHANGE('aaaaaaaa','a',A',0,3,2)      ⟶    'aaAAaaaa'
       /*  all occurrences of old changed, starting at      */
       /*  position 3 for a length of 2                      */
```

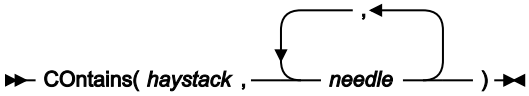
Example 6

```
CH('abcabcabc','a',,0)               ⟶    'bcbcbc'
       /*  new omitted, count = 0,                          */
       /*  all occurrences of old deleted                    */
```

Example 7

```
CHANGE('abc',,'def',,2)              ⟶    'adefbc'
       /*  old omitted, new inserted, starting at           */
       /*  position 2                                        */
```

CONTAINS



If the *haystack* string contains one or more of the *needle* strings, CONTAINS returns 1. Otherwise, CONTAINS returns 0.

CONTAINS is case-sensitive: it only returns 1 if the *haystack* contains a string with the same mix of uppercase and lowercase as a *needle*.

Example 1

If the current input record contains *"Michael"*, *"Mick"* or *"Mike"* in the first ten columns, print the record.

```
If co(fld(1,10),'Michael','Mick','Mike') Then print(inrec, 'CHAR')
```

Example 2

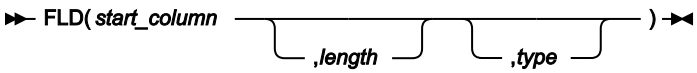
If the current input record contains *"USA"*, *"Australia"* or *"England"*, drop the record from processing.

```
If contains(inrec,'USA','Australia','England') Then exit 'DROP'
```

Related tasks

NCONTAIN on page 274

FLD



Returns the value of a field from the current input record (INREC), starting at *start_column* (in bytes), of *length* number of bytes, interpreted according to the specified *type*:

B

if the field is binary. If you specify B for *type*, *length* must be 2, 4, or 8.

C

if the field contains characters.

P

if the field is packed decimal. If you specify P for *type*, *length* must be between 1 and 16.

Z

if the field is zoned decimal. If you specify Z for *type*, *length* must be between 1 and 32 or, if the field contains a separate sign character, between 1 and 33.

The default value for *type* is C.

If you omit *length* and specify *type* P (packed decimal), FLD attempts to determine the packed field length from the record data, and returns only that field. If you omit *length* for other field types, FLD returns the remainder of the record.

The FLD function is similar to the built-in REXX SUBSTR function, except that FLD interprets the *"substring"* according to the specified data type, and returns the value formatted appropriately. (For a numeric field, FLD returns the value with a sign, and without leading zeros.)

Example 1

If the value of the packed-decimal field that starts at byte 8 is greater than 100, do not process the current record.

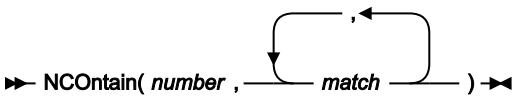
```
 if fld(8,P) > 100 then exit 'DROP'
```

Example 2

If the value of the 2-digit year field starting at byte 42 is greater than 60, insert the literal *"19"* before the year field. Otherwise, insert *"20"*.

```
if fld(42,2,Z) > 60 then
   outrec = fld(1,41)||'19'||fld(42)
else
   outrec = fld(1,41)||'20'||fld(42)
```

NCONTAIN



If the numeric value of any of the *match* arguments is equal to the numeric value of *number*, NCONTAIN returns 1. Otherwise, NCONTAIN returns 0.

Example 1

If the current record contains a packed decimal value of value of 10, 20, or 30 starting at byte 8, print the record.

```
if nco(fld(8,P),10,20,30) then print(inrec,'CHAR')
```

Example 2

If the current record contains a zoned decimal value of 11, 12, or 13 starting at byte 10, drop the record from processing.

```
if nco(fld(10,5,Z),11,12,13) then exit 'DROP'
```

Related tasks

CONTAINS on page 272

PRINT



Prints the *record* string in the specified *format*: SNGL or TABL.

Note:

1. Only those fields that have been selected in the template are printed.
2. If the record has been reformatted by template processing, the variable INREC contains the input record value and OUTREC will contain the reformatted output record.

Example 1

Print the first hundred records.

```
if PRTCOUNT() < 100 then PRINT(inrec,'CHAR')
```

Example 2

Print the current input record in TABL format.

```
rc = PRINT(inrec,'TABL')
```

Related tasks

Working with templates on page 74
Selecting a display format on page 153

PRTCOUNT



Returns the current count of records printed.

Example

Print the first 10 input records.

```
if prtcount() < 10 then print(inrec,'CHAR')
```

RECSIN



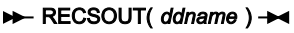
Returns the count of records read so far from the input data set, and passed to the REXX procedure. Only the input records that are selected by the template are passed to the REXX procedure and included in the count returned by RECSIN.

Example

Print every hundredth record.

```
if RECSIN()//100 = 0 then PRINT(inrec,'CHAR')
```

RECSOUT



Returns the count of records so far written to the specified output data set.

The argument you can specify is:

ddname

Specifies that the count of records so far written to the data set identified by the specified *ddname* be returned.

The count of records written to an output data set is incremented each time a WRITE function is issued against the specified data set. In the case of the primary output data set, the count is also incremented each time a record is written to the data set by the ZDT/Db2 function. Unless a record is discarded using the RETURN DROP (or STOP IMMEDIATE) instruction, each record selected for processing is written to the primary output data set.

Example 1

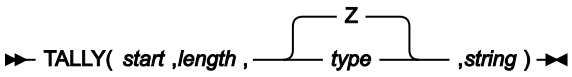
If more than one hundred records have been written to the EXT100 file, stop ZDT/Db2 processing.

```
rc = WRITE(EXT100)
if RECSOUT(EXT100) > 100 then exit "STOP"
```

Related tasks

RETURN values on page 278

TALLY



Accumulates the value of the specified input record field in a TALLY register and, at the end of the Import utility, prints on SYSPRINT the TALLY register prefixed by the character

string *string*. If the TALLY function is successful, it returns a value of 0. If it is unsuccessful, it raises the REXX syntax error condition.

The field whose value is to be accumulated starts at position *start* (in bytes) in the input record, and is *length* bytes long. If the sum of *start* and *length* is greater than LENGTH(INREC), the TALLY function returns a value of 0 without changing the TALLY register.

The data type of the field to be accumulated is specified by *type*. The values that can be specified for *type* are:

B

if the field is binary. If you specify B for *type*, *length* must be 2, 4, or 8. The field is assumed to be signed.

P

if the field is packed decimal. If you specify P for *type*, *length* must be between 1 and 16 bytes.

Z

if the field is zoned decimal. If you specify Z for *type*, *length* must be between 1 and 32 bytes or, if the field contains a separate sign character, between 1 and 33 bytes.

The default value for *type* is Z.

You can code more than one TALLY function in your procedure. ZDT/Db2 creates a separate TALLY register for each TALLY function with a unique combination of arguments. This means that you can accumulate a given field in more than one TALLY register by specifying a different value for *string* in each TALLY function.

Example

Accumulate hours recorded in personnel records depending on record type.

```
select;
  when(fld(1,1) = 'E') then
```

```
    rc = TALLY(15,4,B,'Sum of employee hours')
 when(fld(1,1) = 'S') then
    rc = TALLY(15,4,B,'Sum of supervisor hours')
 otherwise
    rc = TALLY(28,4,B,'Sum of manager hours')
end
```

TM



Tests selected bits of a string and sets the condition code accordingly.

The length of the test is based on the length of the shorter of the two arguments, *string* and *mask*. A *mask* bit of one indicates that the equivalent bit in *string* is to be tested. If a *mask* bit is zero, the equivalent *string* bit is ignored. If the tested bits are all ones, TM returns 1. Otherwise, TM returns 0.

Example 1

Test the third byte of the input record and if the low order bit is set, overlay a hex FF into the second byte of that record.

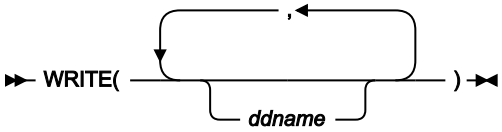
```
if TM(fld(3,1),'01'x) then do
  outrec = OVERLAY('FF'x,outrec,2)
  exit
 end
exit drop
```

Example 2

Test the third byte of the input record and if the high order bit is set, logically OR a hex 04 over the contents of the second byte of that record.

```
if TM(fld(3,1),'10000000'b) then do
  outrec = OVERLAY(BITOR(fld(2,1),'04'x),outrec,2)
  exit
end
exit drop
```

WRITE



Writes a record to the specified data sets. If the WRITE function is successful, it returns a value of 0. If the WRITE function is unsuccessful, it raises the REXX syntax error condition.

The argument you can specify is:

ddname

Specifies a record is to be written to the data set identified by the specified *ddname*.

The record is written from the contents of one of the following REXX variables:

- If it has been assigned a value by the procedure, the variable OUTREC.*ddname*, where *ddname* is the ddname specified in the WRITE function
- If the variable OUTREC.*ddname* has not been assigned a value, or has been unassigned using the DROP instruction, the ZDT/Db2-defined variable, OUTREC

The length of the record written depends upon the data set attributes of the output data set. If the output data set contains variable-length records, the length of the record is determined from the length of the data in the REXX variable. If the length of the data is greater than the maximum record length specified in the data set attributes, the record is truncated. If the output data set contains fixed-length records, the length of the record written is the length specified in the data set attributes, truncated or padded as necessary.

If the record format of the output data set specifies that the records contain a carriage control character, depending on the output device, the first character of the record data in the REXX variable is interpreted as a carriage control character. For more information about records containing carriage control characters, see *OS/390 DFSMS: Using Data Sets*.

Example 1

If the current record is type 01, write it to the DD01 file.

```
if fld(1,2) = 01 then WRITE('DD01')
```

Example 2

If the current record is type 02, write it to the DD02 and DD02COPY files.

```
if fld(1,2) = 02 then WRITE('DD02','DD02COPY')
```

RETURN values

In REXX, you can use the RETURN instruction to leave a procedure. You can optionally specify a character string as a parameter on the RETURN instruction. This character string is returned to the caller of the procedure. Z Data Tools recognizes and processes the following character strings if specified on the RETURN instruction. The character strings can be specified in uppercase, lowercase, or a mixture of both.

DROP

The character string DROP tells ZDT/Db2 to not write the current record to the primary output data set. Specifying DROP on the RETURN instruction results in the current record not being imported to the target table.

Here are some examples:

```
RETURN 'DROP'
Return 'Drop'
Return drop  /* assumes no variable named 'drop' has been assigned */
```

STOP

The character string STOP tells ZDT/Db2 to stop the current ZDT/Db2 function after the current record has been written to the primary output data set. Specifying STOP on the RETURN instruction results in the current record being imported to the target table and the Import utility being stopped.

Here are some examples:

```
RETURN 'STOP'
Return 'Stop'
Return stop  /* assumes no variable named 'stop' has been assigned */
```

STOP IMMEDIATE

The character string STOP IMMEDIATE tells ZDT/Db2 to stop the current ZDT/Db2 function without writing the current record to the primary output data set. Specifying STOP IMMEDIATE on the RETURN instruction results in the current record not being imported to the target table and the Import utility being stopped.

Here are some examples:

```
RETURN 'STOP IMMEDIATE'
Return 'Stop Immediate'
Return stop immediate  /* assumes variables 'stop' and 'immediate' not assigned */
```

Copying data to a VSAM or QSAM file

To copy data from a Db2® table or view to a partitioned, sequential, or VSAM data set, use the Export utility function (3.7).

Note:

1. To copy data between Db2® tables, use the Copy utility function (3.3).
2. To copy data to a Db2® table or view from a QSAM or VSAM data set, use the Import utility function (3.6).
3. To copy data to a sequential file using the Db2® UNLOAD utility, use the Db2® Utilities option (3.9). When copying large tables, the use of the Db2® UNLOAD utility is preferable (for reasons of performance and efficiency).

The Export utility generates the export data using Db2® SQL. The data can be exported in any of these formats:

- Db2® (Z Data Tools internal) format. This format is *not* the same as the data formats used by the DSNTIAUL sample job and the Db2® utilities. This is the recommended option if the exported data is imported into another Db2® table using the ZDT/Db2 Import utility.
- Db2® external (or unload) format. This is the same format used by the Db2® UNLOAD utility.
- Db2® DSNTIAUL unload format. This is the format used by the Db2® DSNTIAUL sample program.
- A user defined format. Note that the user defined Placement: **3. User defined** is not supported by the Import utility.
- Delimited (CSV) format. The exported data format is not supported by the Import utility.
- Copybook layout format.

You can use the output data set generated by the Export Utility for input to:

1. The ZDT/Db2 Import utility. This utility supports data in Db2® format and copybook formats.
2. A PC-based spreadsheet application program in the form of a comma-delimited (CSV) file.
3. Other user application programs.

If you export data, you can:

- Select the format in which the data is exported.
- Select the columns to be exported by selecting the required columns in the "from" template.
- Reformat data during export by mapping table columns to fields in the output partitioned, sequential, or VSAM data set (as defined by a template).
- Create and save a template that describes the exported data.
- Select which rows to export.
- Specify the number of rows that are exported.
- Specify whether you want to run the export in foreground or batch.

To display the Export Utility panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `7` (Export)

Related tasks

Copying data from a VSAM or QSAM file on page 251
Notes on using Export on page 287
Specifying details for the "From" table on page 281
Specifying details for the "To" data set on page 284

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Export Utility ("From") panel on page 698

Specifying details for the "From" table

Specify the name of the source (*"From"*) table on the Export Utility panel. If the table you specify does not exist, ZDT/Db2 issues an error message and stops the utility.

If you do not want to export all of the rows in the *"From"* table, specify the number of rows you want to export (1–99999999) in **Export Count**.

If you are using a specific template for the *"From"* table (see option 1 in **Template usage**), specify the name of the template in the **From Template** entry fields.

The **Processing Options** on the Export Utility panel specify which template you want to use for the *"From"* table, and whether you want to edit the template before the export process.

Template usage

The Export Utility panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **From Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (see Template usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **From Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. **Generate/Replace**

   Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **From Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a **"/"** or an "A":

Edit options

Use to display the first of the Export Options panels.

Edit template

Use to edit the template for the *"From"* table before proceeding to the Export Utility *"To"* panel.

To proceed, press Enter.

If you selected **Edit options** (either by using a "/", or by having previously used the *"A"* selection character for this option), the first of the Export Options panels is displayed.

If you selected **Edit template** (either by using a "/", or by having previously used the *"A"* selection character for this option), the Column Selection/Edit panel is displayed.

Otherwise, the Export Utility *"To"* panel is displayed.

Related tasks

Specifying a Db2 object name on page 46
Specifying a data set and a member name on page 52
Selecting options on ZDT/Db2 panels on page 45
Working with templates on page 74
Editing the template for the "From" table on page 283
Specifying details for the "To" data set on page 284

Related references

Export Utility ("From") panel on page 698

Setting options for the current Export session

If you select **Edit options** on the Export Utility panel (or the Export Utility *"To"* panel), ZDT/Db2 displays the first of up to three Export Options panels. The remaining panels are displayed by pressing the NxtPage function key (F11).

Note:

1. To navigate to the Export Options (2 of 3) panel, you must select one of the data format options **4. User defined** *or* **5. Delimited variables (CSV)**, *and* **Null indicators / CSV options** on the Export Options (1 of 3) panel.
2. To navigate to the Export Options (3 of 3) panel, you must select the data format option **4. User defined** *and* **Data type format** on the Export Options (1 of 3) panel.

Use the Export Options panels to change the options for the current Export session.

Note: Options apply for the current session of ZDT/Db2 and only revert to global settings when the session is terminated or global settings are invoked (from the pull down or from the menu).

Related references

Export Utility ("From") panel on page 698
Export Options (1 of 3) panel on page 689

Editing the template for the "From" table

If you select the **Edit template** option on the Export Utility panel, ZDT/Db2 displays the Column Selection/Edit panel. You can use this panel to restrict the data in the *"From"* table that you want exported by:

- Selecting (or deselecting) specific columns for exporting.
- Specifying *row selection criteria*. Only rows whose contents satisfy the criteria are selected for exporting.

Note:
When you export data in ZDT/Db2 internal (SQLDA) format, you cannot alter the order in which the columns are exported. Data is always exported in the order that the columns are defined in the Db2® catalog (the same order in which the columns are displayed in the Column/Selection Edit panel).

The Export Utility ignores any information specified in the **Seq** field of the template (this information is used to change the order in which ZDT/Db2 *displays* or *prints* columns).

For example, the template shown in Figure 66: Column Selection/Edit panel for the “From” table on page 283 has been edited to select certain columns and only rows whose contents meet certain criteria. If you use this template as the *"From"* template for your Export, the data exported is restricted to:

- The columns FIRSTNME, LASTNAME, WORKDEPT, and JOB, but only where WORKDEPT equals *"MNT"* and JOB equals *"PAINTER"*.

Figure 66. Column Selection/Edit panel for the “From” table

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)              Column Selection/Edit                Line 1 of 16
TABLE HFMUSER.EMP
------- Row Selection Criteria ---- (Use SQL/PF4 for full screen edit) -------
1 Sel: WHERE "WORKDEPT" = 'MNT' AND "JOB" = 'PAINTER'                      +

Cmd Seq SHEX CL# Column name          Data type(length)  Null Default  Order A/D
                ****  Top of data  ****
___ ___          1 EMPNO              CHARACTER(6)            None       ___   _
___ ___ S        2 FIRSTNME           VARCHAR(12)             None       ___   _
___ ___          3 MIDINIT            CHARACTER(1)            None       ___   _
___ ___ S        4 LASTNAME           VARCHAR(15)             None       ___   _
___ ___ S        5 WORKDEPT           CHARACTER(3)        Y   Null       ___   _
___ ___          6 PHONENO            CHARACTER(4)        Y   Null       ___   _
___ ___          7 HIREDATE           DATE                Y   Null       ___   _
___ ___ S        8 JOB                CHARACTER(8)        Y   Null       ___   _
___ ___          9 EDLEVEL            SMALLINT            Y   Null       ___   _
___ ___         10 SEX                CHARACTER(1)        Y   Null       ___   _
___ ___         11 BIRTHDATE          DATE                Y   Null       ___   _
___ ___         12 SALARY             DECIMAL(9,2)        Y   Null       ___   _
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=SQL       F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Working with templates on page 74
Changing the sequence of displayed or printed data on page 101

Related references

Export Utility ("From") panel on page 698
Column Selection/Edit panel on page 497

Specifying details for the "To" data set

Type the "To" data set details. Use a combination of the **Data set name**, **Member**, and **Volume** entry fields to specify the partitioned, sequential, or VSAM data set to which you want to export the data. You can specify the name of an existing data set, or the name of a data set that you want the Export utility function to create.

If you are using a specific template or copybook for the *"To"* data set, specify the name of the template or copybook data set and, optionally, the member name, in the **To Copybook or Template** entry fields.

Use the **Processing Options** to select a number of options for the export process.

To proceed, press Enter.

If you selected **Edit options** (either by using a "/", or by having previously used the *"A"* selection character for this option), the second of the Export Options panels is displayed.

If you selected **Edit template mapping** (either by using a "/", or by having previously used the *"A"* selection character for this option), the Template Mapping panel is displayed for editing.

If you selected **Batch execution** (either by using a "/", or by having previously used the *"A"* selection character for this option), the generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

If you selected options **3. Batch, using Db2® utility** and **Edit UNLOAD options** on the Export Options - (1 of 3) panel, before the generated JCL is displayed using the ISPF editor, the UNLOAD Options panel is displayed.

Otherwise, ZDT/Db2 exports the data from the *"From"* table to the *"To"* partitioned, sequential, or VSAM data set.

Related tasks

Specifying a data set and a member name on page 52
Mapping data on page 112
Exported data formats on page 285
Running Export in batch on page 286
Notes on using Export on page 287

Related references

Export Utility ("To") panel on page 701
Template Mapping panel on page 900

Exported data formats

You can export data in four pre-defined formats:

- Db2® (Z Data Tools internal) format
- Db2® external (or unload) format
- Db2® DSNTIAUL unload format
- Comma-delimited format (CSV output)

and two formats that require user definition:

- A user-defined format
- Copybook layout format

The output format depends on which **Data format** option you selected on the Export Options - (2 of 3) panel:

To export data in Db2® (Z Data Tools internal) format, specify option 1 (**ZDT/Db2 (SQLDA) format**).

To export data in Db2® external (or unload) format, specify option 2 (**Db2® unload format**).

To export data in Db2® DSNTIAUL unload format, specify option 3 (**DSNTIAUL format**).

To export data in user-defined format, specify option 4 (**User defined**).

To export data in comma-delimited format (CSV output), specify option 5 (**Delimited Variables (CSV)**).

The output format depends on which **Template usage** processing option you select.

To export data in Db2® (Z Data Tools internal) format, specify **Template usage** option 3 (**Generate from input**) or option 4 (**Generate and save**) and export option 1 **ZDT/Db2 (SQLDA) format**. When you specify these options, ZDT/Db2 copies the output template from the input template, and exports the data in the format returned by Db2® if an SQLDA is used to access the result table. In this format, any nullable column is preceded by a 2-byte binary field containing the null indicator. If the *"From"* column has a null value, the null indicator field for the corresponding *"To"* field contains -1, otherwise it contains 0. Each decimal field in the output data set occupies 16 bytes, with the data left-aligned followed by trailing binary zeros. Any graphic or vargraphic field is followed by two bytes containing binary zeros. Use this option if you plan to import the exported data into another Db2® table using Import (option 3.6).

To export data in Db2® unload format, specify Template usage option 3 (**Generate from input**) or option 4 (**Generate and save**) and select export option 2 **Db2® unload format**. The exported data is in the same format as that returned by the Db2® UNLOAD utility, with the NOPAD option not specified. In this format, any nullable column is preceded by a 1-byte binary field containing the null indicator. If the “From” column has a null value, the null indicator field for the corresponding “To” field contains X'FF', otherwise it contains 0. Any decimal field occupies the minimum number of bytes needed to represent the data as a packed decimal field. Variable length columns in the unloaded records are padded to their maximum length and the padded data fields are preceded by length fields that indicate the size of the actual data without the padding. CCSID conversion is not supported. The exported data will be in the CCSID used to bind the ZDT/Db2 plan.

To export data in DSNTIAUL format, specify Template usage option 3 (**Generate from input**) or option 4 (**Generate and save**), and select export option **3. DSNTIAUL format**. In this format, any nullable column is followed by a 1-byte binary field containing the null indicator. If the "From" column has a null value, the null indicator field for the corresponding "To" field contains "?", otherwise it contains 0. Any decimal field occupies the minimum number of bytes needed to represent the data as a packed decimal field.

To export data in comma-delimited format (CSV output), specify **Template usage** option 5 (**None. CSV output**). If you specify this option, ZDT/Db2 does not generate an output template but writes the exported data as varying-length records, separated by a delimiter. For delimited format, ZDT/Db2 uses the **Null column display indicator** to indicate that data for a nullable column is the null value.

To export data in user-defined format, specify **Template usage** option 3 (**Generate from input**) or option 4 (**Generate and save**), and select export option **4. User defined format**. For a user-defined format, ZDT/Db2 provides the following options for nullable columns:

- To use or not use nulls. In the latter case, there would be no fields in the exported data corresponding to the null indicator. Set Null Indicators Usage to **2. None** on Export Options screen.
- To place the null indicator either before or after the data for the column to which it applies. Set Null Indicators Placement to **1. Before** or **2. After** on the Export Options screen.
- To use either a 1-byte (character format) or 2-byte (signed integer format) for the null indicator, set Null Indicators Type to **1. One byte** or **2. Two byte** on the Export Options screen. Set the value to be used in Indicator **Char** or **Integer**.

To export data in user-defined format, and to place either a 1-byte (character format) or a 2-byte (signed integer format) null indicator somewhere in the output record, but not before or after the data for the column to which it applies, select Null Indicators Placement to **3. User defined** on the Export Options - (2 of 3) panel. You must provide a copybook and specify Template usage option **1 (Above)** or **2 (Previous)**, and select **Edit template mapping** and map the null indicators. If the field names in the copybook are not the same as the Db2® column names, you will also need to map each Db2® column to the appropriate field in the copybook.

To export data in any other format, you must use a user-defined template. Specify **Template usage** option 1 (**Above**) to use the template you have specified in the **To Copybook or Template** section of the panel, or **Template usage** option 2 (**Previous**) to use the template last used for the table from which you are exporting data. If exporting columns that permit null values, ZDT/Db2 processes them as follows: if the field that corresponds to a nullable column is followed by an unmapped 2-byte binary field, ZDT/Db2 maps the null indicator for the column to this 2-byte binary field. In all other situations, ZDT/Db2 does not map the null indicator to any field in the output template.

Related references

Editor Options (4 of 8) panel on page 662

Running Export in batch

If you want to export large Db2® tables, you can perform the export in a batch job.

If you select the **Batch execution** option on the Export Options (1 of 3) panel, the Export utility builds the necessary JCL to perform the export function in a batch job. The generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing. If you select the **Batch, using Db2® utility** option, the JCL is set up for the Db2® UNLOAD utility, and displayed for review before submission.

If you have one or more data sets that contain the template that describes the record structure of your input data, you can specify the data set details (or TSO ALLOC statement) in the generated JCL with the keywords TINPUT or TCIN.

Similarly, if you have one or more data sets that contain the template that describes the record structure of your output data, you can specify the data set details (or TSO ALLOC statement) in the generated JCL with the keywords TOUTPUT or TCOUT.

If the export data set does not already exist, you are prompted to create it. However, you can defer the creation of the export data set until batch job execution time by selecting the **Batch data set creation** option. If you select this option, ZDT/Db2 creates a new sequential export data set in the JCL for the batch job.

Related references

DBX (Export) batch command on page 1043

Notes on using Export

The Export utility function provides one method for moving Db2® data between different Db2® sub-systems (that is, to export data from one Db2® sub-system and import it to another Db2® sub-system). If you want to move data between tables in the same Db2® sub-system, you might find it more convenient to use Copy (3.3), or move the data by direct execution of SQL statements.

In general terms, the Export utility function is suitable for extracting small to medium volumes of data from Db2® tables. It is not intended to serve as a substitute for the Db2® utilities if unloading or migrating large volumes of data. The Db2® utilities offer superior performance in these situations.

It is recommended that you run the Export utility in batch whenever possible.

The Export utility provides for automatic mapping of the Db2® null indicator. However, if Export Option Data format 4. User defined Null Indicators Placement 3. User defined is selected the template editor shows the null indicator as a separate item in the “From” template on the Field Mapping panel. It is therefore possible to re-map the null indicator associated with nullable columns to a field in the export data set. If you want to rearrange the data fields in the export data set, it is recommended that you first export the data using the default mapping and then use the Z Data Tools Base component to map the data to the required format.

A sample REXX exec is provided in SHFMSAM1 that converts a ZDT/Db2 template generated by ZDT/Db2 into a COBOL or PL/I copybook.

To convert a ZDT/Db2 template generated by ZDT/Db2 into a COBOL or PL/I copybook, you specify a TSO command like the following:

```
HFM2CPYB input output language
```

where:

input

The name of a sequential file or partitioned data set and member that contains a Z Data Tools Db2® template.

output

The name of a sequential file or partitioned data set and member that will contain the copybook.

language

Optional keyword that determines the type of copybook generated. Valid values are:

COBOL
PL/I

If not specified, a COBOL copybook is generated.

The sample REXX must be copied to a library in your SYSPROC or SYSEXEC concatenation. Alternatively, you can specify the library name containing the sample exec in the command.

Related tasks

Copying data from one Db2 object to another on page 246
Generating batch JCL for Db2 utility jobs on page 346
Working with templates on page 74

Copying data from a Z Data Tools/Db2 editor session, or from an SQL statement

Use the EXPORT primary command to write data from the current ZDT/Db2 editor session to an external file. Alternatively, you can re-execute the SQL statement that generated the data for the current ZDT/Db2 editor session and write rows for this select statement to an external file. For the second option, you can generate a JCL deck to perform the export operation using batch processing.

Data can be exported in the same data formats as for the Export utility (3.7).

During the export process, the Export Utility panel is optionally displayed. You enter the data format for the exported data on this panel.

The Export "To" panel is also optionally displayed. The Export "To" panel enables you to:

- Specify the name of the data set that will contain the exported data.
- Specify the name of a data set that contains a template or copybook, that describes the exported data.
- Specify a processing option for the template data set.
- Specify a disposition option for the export data set.
- View and possibly change the export options.
- Use the template editor to display and possibly change the mapping between the source and data to be exported.

Related references

- Export Utility ("From") panel on page 698
- Export Utility ("To") panel on page 701
- Specifying details for the "From" data set on page 253
- Specifying details for the "To" table on page 249
- EXPORT primary command on page 950

Specifying the source for the exported data

You can specify the source for the exported data with the ALL, NX, X, or EX parameter to the EXPORT primary command.

When you specify this parameter, ZDT/Db2 copies all, or selected, rows in the current ZDT/Db2 editor session to the target data set.

If you omit this parameter, ZDT/Db2 re-executes the SQL statement that produced the data for the current ZDT/Db2 editor session. ZDT/Db2 exports the rows fetched by the SQL statement to the target data set, up to any row limit specified on the export options panel.

You can use the **Export Count** field on the Export Utility panel to limit the data loaded into the ZDT/Db2 editor, for large Db2® objects, or arbitrary select statements that generate large result tables. Issuing the EXPORT primary command without the ALL parameter enables all rows for the Db2® object, or result table from the arbitrary select statement, to be written to the export data set.

Related references

- EXPORT primary command on page 950
- Export Utility ("From") panel on page 698

Specifying the format of the exported data

You can export data in a number of data formats (see Exported data formats on page 285).

You can specify the data format for the exported data as an optional parameter to the EXPORT primary command. When specified, the parameter overrides the current data format, as displayed on the Export Options (1 of 3) panel.

The possible parameter values are SQLDA, Db2®, DSNTIAUL, USER, and CSV.

Related references

- EXPORT primary command on page 950
- Export Options (1 of 3) panel on page 689

Limiting the rows exported with a label range

You can limit the rows processed to only those rows within a specified label range, rather than all rows in the ZDT/Db2 editor session.

Note: You can only specify a label range if you have also specified the source parameter (ALL, NX, X, or EX).

You must specify a "from" label and a "to" label.

Related references

Assigning labels to rows on page 150
EXPORT primary command on page 950

Differences compared with the export utility

The process for exporting data with the EXPORT primary command is similar to that for the Export utility function (3.7) with the following exceptions:

- The **Batch, using Db2® UNLOAD** execution option is not available when exporting data using the EXPORT primary command.
- The dialog commences with the optional display of the Export Options (1 of 3) panel. The Export Utility panel is not displayed.
- The data to be exported can be the result table for the SQL statement that generated the current ZDT/Db2 editor session, *or* rows from the current ZDT/Db2 editor session. In the former case, the SQL statement can refer to a Db2® object (for example, when entered using the Edit function), or an arbitrary SQL statement (for example, when entered using one of the SQL options 4.1, 4.2, 4.3 or 4.4). The arbitrary SQL statement can be any valid select statement, including joins of multiple tables, sub-selects, and so on. By contrast, the Export utility only allows export of data from a Db2® object (for example, table, view, alias).
- When using the EXPORT primary command to export data from an arbitrary SELECT statement (for example, entered using one of the ZDT/Db2 functions 4.1, 4.2, 4.3 or 4.4) in batch, there are some restrictions on the template mapping operations that are supported.

  It is not possible to use the template used to describe the SELECT statement in the editor session in the generated JCL deck.

  For export in CSV format, which cannot be described using a template, this means any changes made to the template for the editor session are discarded when exporting the data - the output will include data for every column.

  For export in other supported formats:

  - Use the TE command in the editor session to select or de-select the columns as required.
  - Issue the EXPORT BAT command to generate a JCL deck.
  - You are required to save the output template that describes the data to be exported. This is included as a parameter in the JCL deck.
  - When the JCL deck is submitted for processing, ZDT/Db2 re-creates a temporary template for the arbitrary SQL statement specified in the JCL deck, and then uses template mapping to identify which columns should be exported - these are the columns defined in the output template.

- The export options panel includes an input field where the number of rows to be exported can be entered. This operates independently of any input field that limits the number of rows loaded into the ZDT/Db2 editor session. It is therefore possible to edit a few rows of a result table using ZDT/Db2 edit, and then export *all* the rows for that result table to a data set.

Related references

Export Options (1 of 3) panel on page 689
Export Utility ("From") panel on page 698
EXPORT primary command on page 950

Ways that you can use the EXPORT primary command

Here are two ways you can use the EXPORT primary command:

- To export data from an arbitrary SQL statement to an external file:
  1. Develop the SQL SELECT statement using options 4.1, 4.2, 4.3, or 4.4. The SQL statement can be any valid SQL statement, including joins of multiple tables, sub-selects, and so on.
  2. Use the row count field to limit the rows loaded into the ZDT/Db2 editor, if the result table is large.
  3. Confirm that the result table is correct by examining the data in the Edit session.
  4. Optionally edit the template and de-select any columns of the result table that are not required.
  5. Issue the EXPORT command without the ALL parameter, to export all rows from the result table of the arbitrary SQL statement to an external file. You can also issue EXPORT BAT to generate a JCL deck to export the rows from the arbitrary SQL statement to an external file. If required you can take a copy of the generated JCL for later re-use.
- To export selected editor session rows to an external file:
  1. Ensure that the **Use edit (instead of browse) to display results** option is selected. This option is set using the ZDT/Db2 System Options (2 of 3) panel.
  2. Develop the SQL SELECT statement using options 4.1, 4.2, 4.3 or 4.4, or edit the required Db2® object using the ZDT/Db2 Edit utility.
  3. Use editor commands to isolate the required data in either excluded, not excluded rows, label ranges, or some combination of these methods.
  4. Issue the EXPORT primary command with the OPT parameter and one of ALL, X, or NX. Ensure that the required data format is selected. On the Export "To" panel, enter the required export data set name and set the disposition option to MOD.
  5. After the initial export of rows to the export data set, isolate additional rows to be exported, then issue EXPORT QUIET plus a combination of ALL, X, or NX and optionally a label range, to export the additional rows. The MOD setting for the export file ensures that each additional EXPORT primary command adds the newly-exported rows to the end of the export data set.

Scrambling data

Scrambling data allows you to create test data based on production (or "live") data, but with the ability to change the values of certain columns. In this way, you can avoid sensitive or confidential information appearing in test data.

When can you scramble data?

When you copy data from one place to another with either the Copy Utility (option 3.3), Import Utility (option 3.6), or Export Utility (option 3.7), you can choose to scramble some or all of the columns being copied.

For scrambling to occur during the copy, import, or export process, you must supply an *output template* which has mapped fields *marked for scrambling*.

Scrambling rules

ZDT/Db2 scrambles data according to these rules:

- If you do not specify any value or range options:
  - Uppercase alphabetic characters are scrambled to other uppercase alphabetic characters.
  - Lowercase alphabetic characters are scrambled to other lowercase alphabetic characters.
  - DBCS characters are scrambled to other DBCS characters.
  - Numerics are scrambled to other numerics.
  - Any other characters remain unchanged.
  - Repeatable scrambling produces unique results for numeric fields defined with **Leading zeros** set to YES, and all non-numeric fields.
- If you provide a value list, then the column is populated with a value from the list.
- If you provide a range, then the resultant number is in the range provided.
- If you specify a scramble type of random or repeatable and also the value option and a value data set, then the column is populated with a value from the data set.
- If you specify a scramble type of translate, then the input column value is matched and the corresponding output value is used from the value data set.

Related topics

Specifying scrambling options on page 108
Column Attributes panel (alphanumeric) on page 484
Column Attributes panel (numeric) on page 490
Value List Edit panel on page 917
Scramble Exit Specification panel on page 814

How do I...?

If you want to copy data from one place to another, but are not sure of the best way of going about it, you may find the following sections helpful.

The sections below describe typical situations that you may encounter when planning to copy data. For each scenario, step-by-step instructions are included describing how to achieve the required result.

Note: In the following examples, *"source table"* refers to the table from which the data is exported, and *"target table"* refers to the table to which the data is imported.

How do I copy data from one Db2® table to another on the same Db2® system?

Scenario: I have data in a Db2® table and want to copy it to another Db2® table. The source and target Db2® tables are on the same Db2® system.

Consider using the ZDT/Db2 Copy Utility (3.3), instead of the ZDT/Db2 Export Utility and Import Utility. If the source and target table have different column names, different numbers of columns, or if data type conversions are needed, you can use template re-mapping to map columns in the source table to columns in the target table. In most cases, ZDT/Db2 performs any data type conversions automatically.

Related tasks

Copying data from one Db2 object to another on page 246
Column mapping rules on page 118

How do I copy data from one Db2® table to another on a different (connectable) Db2® system?

Scenario: I have data in a Db2® table and want to copy it to another Db2® table. The source and target Db2® tables are on different Db2® systems. It is possible to connect to the target Db2® system from the source Db2® system by specifying a location value.

Consider using the ZDT/Db2 Copy Utility (3.3), instead of the ZDT/Db2 Export Utility and the ZDT/Db2 Import Utility. See How do I copy data from one Db2 table to another on the same Db2 system? on page 293. If a large number of rows are to be transferred, some caution needs to be exercised due to network considerations.

Related tasks

Copying data from one Db2 object to another on page 246
Column mapping rules on page 118

How do I copy data from one Db2® table to another on a different (non-connectable) Db2® system?

Scenario: I have data in a Db2® table and want to copy it to another Db2® table. The source and target Db2® tables are on different Db2® systems. It is not possible to connect to the target Db2® system from the source Db2® system by specifying a location value, or I want to transfer the data out of Db2® as part of the copy process.

To copy the data, you need to perform two steps:

1. Use the ZDT/Db2 Export Utility (3.7) to export the data to a sequential file (preferably) or a VSAM file. In general, the default mapping is sufficient. To use the default mapping, specify a template data set and member name on the Export *"To"* panel and select **Template usage** option 4 (Generate and save). If the amount of data to be transferred is large (more than 10,000 rows), run the export in batch by selecting **Batch execution**.

   If the amount of data to be transferred is very large (more than 100,000–1,000,000 rows), consider using the Db2® UNLOAD utility (accessed from Utilities, 3.9) instead.

2. Use the ZDT/Db2 Import Utility (3.6) to load the exported data into the target Db2® table. If the target table has different column names to the source table, or if data type conversions are required, use template re-mapping to map columns in the source table to columns in the target table. In most cases, ZDT/Db2 performs any data type conversions automatically.

Related tasks

Copying data to a VSAM or QSAM file on page 279
Running Export in batch on page 286
Copying data from a VSAM or QSAM file on page 251
UNLOAD utility on page 380
Column mapping rules on page 118

Related references

Export Utility ("To") panel on page 701

How do I copy and reorder selected data from a Db2® table to a sequential file?

Scenario: I have data in a Db2® table and I want to copy it to a sequential file. I only want the data for some of the columns in the Db2® table, and I want the data for those columns reordered in the sequential data set.

The ZDT/Db2 Export Utility (3.7) supports the export of selected columns from a Db2® object. To export selected columns only, edit the template on the Export Utility panel and deselect the columns that are not required.

The ZDT/Db2 Export Utility does *not* support the reordering of columns in the export data set. The selected columns are always exported in the order in which they are defined in the Db2® catalog. There is a simple workaround for this. Create a view for the Db2® object to be exported, specifying the columns in the desired order. Then export data from the view, to achieve the desired result.

Related tasks

Copying data to a VSAM or QSAM file on page 279
Mapping data on page 112

Related references

Export Utility ("From") panel on page 698

How do I define a copybook that describes data exported to a sequential file?

Scenario: I have data in a Db2® table and I've used the ZDT/Db2 Export Utility to copy the data to a sequential file. I need to define a copybook that describes the exported data.

You need to code the copybook in either COBOL, HLASM, or PL/I.

The amount of space required in the export data set for each of the Db2® data types is summarized in the following example:

- Suppose you want to export data from the table created using the following definition:

```
CREATE TABLE SAMPLE.EXPORT_EX
   (COL1   INTEGER         NOT NULL
   ,COL2   SMALLINT        NOT NULL
   ,COL3   FLOAT(21)       NOT NULL
   ,COL4   FLOAT(53)       NOT NULL
   ,COL5   DECIMAL(1,0)    NOT NULL
   ,COL6   DECIMAL(15,7)   NOT NULL
   ,COL7   CHAR(10)        NOT NULL
   ,COL8   VARCHAR(10)     NOT NULL
   ,COL9   DATE            NOT NULL
   ,COL10  TIME            NOT NULL
   ,COL11  TIMESTAMP       NOT NULL
   ,COL12  ROWID  GENERATED ALWAYS NOT NULL
   )
```

  A COBOL copybook that describes the exported data is as follows:

```
 01  EXPORT-EXAMPLE.
* INTEGER DATA TYPE
     10 COL1                 PIC S9(6) USAGE COMP.
* SMALLINT DATA TYPE
     10 COL2                 PIC S9(4) USAGE COMP.
* FLOATING POINT (4 BYTES) DATA TYPE
     10 COL3                 USAGE COMP-1.
* FLOATING POINT (8 BYTES) DATA TYPE
     10 COL4                 USAGE COMP-2.
* PACKED DECIMAL DATA TYPE
     10 COL5                 PIC S9(1)V USAGE COMP-3.
     10 FILLER               PIC X(15).
* PACKED DECIMAL DATA TYPE
     10 COL6                 PIC S9(8)V9(7) USAGE COMP-3.
     10 FILLER               PIC X(8).
* CHARACTER DATA TYPE
     10 COL7                 PIC X(10).
```

```
* VARCHAR DATA TYPE
     10 COL8.
        49 COL8-LEN          PIC S9(4) USAGE COMP.
        49 COL8-TEXT         PIC X(10).
* DATE DATA TYPE
     10 COL9                 PIC X(10).
* TIME DATA TYPE
     10 COL10                PIC X(8).
* TIMESTAMP DATA TYPE
     10 COL11                PIC X(26).
* ROWID DATA TYPE
     10 COL12.
        49 COL12-LEN         PIC S9(4) USAGE COMP.
        49 COL12-TEXT        PIC X(40).
```

In HLASM, the copybook looks like this:

```
EXAMPLE  DSECT
* Integer data type
COL1     DS    F
* Smallint data type
COL2     DS    H
* Floating point (4 bytes) data type
COL3     DS    E
* Floating point (8 bytes) data type
COL4     DS    D
* Packed decimal data type (1 byte)
COL5     DS    PL1
* Packed decimal data type (8 bytes)
COL6     DS    PL8
* Character data type
COL7     DS    CL10
* VARCHAR data type
COL8_LEN DS    H
COL8_TXT DS    CL10
* DATE data type
COL9     DS    CL10
* TIME data type
COL10    DS    CL8
* TIMESTAMP data type
COL11    DS    CL26
* ROWID data type
COL12_LEN DS   H
COL12_TXT DS   CL40
         END
```

In PL/I, the copybook looks like this:

```
 DCL 1  EXPORT_EXAMPLE UNALIGNED,
/* INTEGER DATA TYPE                                          */
     2 COL1                 BIN(31,0),
/* SMALLINT DATA TYPE                                         */
     2 COL2                 BIN(15,0),
/* FLOATING POINT (4 BYTES) DATA TYPE                         */
     2 COL3                 FLOAT,
/* FLOATING POINT (8 BYTES) DATA TYPE                         */
     2 COL4                 FLOAT(8),
/* PACKED DECIMAL DATA TYPE                                   */
     2 COL5                 FIXED DEC(1,0),
```

```
     2 COL5_PAD              CHAR(15),
/* PACKED DECIMAL DATA TYPE                                    */
     2 COL6                  FIXED DEC(15,7),
     2 COL6_PAD              CHAR(8),
/* CHARACTER DATA TYPE                                         */
     2 COL7                  CHAR(10),
/* VARCHAR DATA TYPE                                           */
     2 COL8,
        3 COL8_LEN           BIN(15,0),
        3 COL8_TEXT          CHAR(10),
/* DATE DATA TYPE                                              */
     2 COL9                  CHAR(10),
/* TIME DATA TYPE                                              */
     2 COL10                 CHAR(8),
/* TIMESTAMP DATA TYPE                                         */
     2 COL11                 CHAR(26),
/* ROWID DATA TYPE                                             */
     2 COL12,
        3 COL12_LEN          BIN(15,0),
        3 COL12_TEXT         CHAR(40);
```

Note the following:

Db2® data type

Space in exported data set.

INTEGER

4 bytes.

SMALLINT

2 bytes.

REAL (4 bytes)

4 bytes.

FLOAT (8 bytes)

8 bytes.

PACKED DECIMAL

16 bytes in total. The data is left-aligned. The unused bytes contain binary zeros. A DECIMAL(*n*,*m*) field occupies TRUNC(*n*/2) + 1 bytes within the 16-byte field.

CHAR(*n*)

n bytes.

VARCHAR(*n*)

n+2 bytes. The data is preceded by a 2-byte binary field, containing the number of characters in the VARCHAR field.

DATE

10 bytes. See note 1.

TIME

8 bytes. See note 1.

TIMESTAMP

26 bytes.

ROWID

42 bytes. Define a ROWID column as for a VARCHAR(40) column.

Note:

1. The length required for DATE and TIME fields may vary if a date or time exit is in use.

- If the Db2® table has columns that allow nulls and was created using the following definition:

```
CREATE TABLE SAMPLE.EXPORT_EX_NULL
   (COL1   INTEGER
   ,COL2   SMALLINT
   ,COL3   FLOAT(21)
   ,COL4   FLOAT(53)
   ,COL5   DECIMAL(1,0)
   ,COL6   DECIMAL(15,7)
   ,COL7   CHAR(10)
   ,COL8   VARCHAR(10)
   ,COL9   DATE
   ,COL10  TIME
   ,COL11  TIMESTAMP
   ,COL12  ROWID  GENERATED ALWAYS NOT NULL
   )
```

A COBOL copybook that describes the exported data is as follows:

```
 01  EXPORT-EXAMPL2.
* INTEGER DATA TYPE WITH NULL INDICATOR
     10 COL1-NULL            PIC S9(4) USAGE COMP.
     10 COL1                 PIC S9(6) USAGE COMP.
* SMALLINT DATA TYPE WITH NULL INDICATOR
     10 COL2-NULL            PIC S9(4) USAGE COMP.
     10 COL2                 PIC S9(4) USAGE COMP.
* FLOATING POINT (4 BYTES) DATA TYPE WITH NULL INDICATOR
     10 COL3-NULL            PIC S9(4) USAGE COMP.
     10 COL3                 USAGE COMP-1.
* FLOATING POINT (8 BYTES) DATA TYPE WITH NULL INDICATOR
     10 COL4-NULL            PIC S9(4) USAGE COMP.
     10 COL4                 USAGE COMP-2.
* PACKED DECIMAL DATA TYPE WITH NULL INDICATOR
     10 COL5-NULL            PIC S9(4) USAGE COMP.
     10 COL5                 PIC S9(1)V USAGE COMP-3.
     10 FILLER               PIC X(15).
* PACKED DECIMAL DATA TYPE WITH NULL INDICATOR
     10 COL6-NULL            PIC S9(4) USAGE COMP.
     10 COL6                 PIC S9(8)V9(7) USAGE COMP-3.
     10 FILLER               PIC X(8).
* CHARACTER DATA TYPE WITH NULL INDICATOR
     10 COL7-NULL            PIC S9(4) USAGE COMP.
```

```
     10 COL7                 PIC X(10).
* VARCHAR DATA TYPE WITH NULL INDICATOR
     10 COL8-NULL            PIC S9(4) USAGE COMP.
     10 COL8.
        49 COL8-LEN          PIC S9(4) USAGE COMP.
        49 COL8-TEXT         PIC X(10).
* DATE DATA TYPE WITH NULL INDICATOR
     10 COL9-NULL            PIC S9(4) USAGE COMP.
     10 COL9                 PIC X(10).
* TIME DATA TYPE WITH NULL INDICATOR
     10 COL10-NULL           PIC S9(4) USAGE COMP.
     10 COL10                PIC X(8).
* TIMESTAMP DATA TYPE WITH NULL INDICATOR
     10 COL11-NULL           PIC S9(4) USAGE COMP.
     10 COL11                PIC X(26).
* ROWID DATA TYPE
     10 COL12.
        49 COL12-LEN         PIC S9(4) USAGE COMP.
        49 COL12-TEXT        PIC X(40).
```

In HLASM, the copybook looks like this:

```
EXAMPLE2  DSECT
* Integer data type with null indicator
COL1_NI  DS    H
COL1     DS    F
* Smallint data type with null indicator
COL2_NI  DS    H
COL2     DS    H
* Floating point (4 bytes) data type with null indicator
COL3_NI  DS    H
COL3     DS    E
* Floating point (8 bytes) data type with null indicator
COL4_NI  DS    H
COL4     DS    D
* Packed decimal data type (1 byte) with null indicator
COL5_NI  DS    H
COL5     DS    PL1
* Packed decimal data type (8 bytes) with null indicator
COL6_NI  DS    H
COL6     DS    PL8
* Character data type with null indicator
COL7_NI  DS    H
COL7     DS    CL10
* VARCHAR data type with null indicator
COL8_NI  DS    H
COL8_LEN DS    H
COL8_TXT DS    CL10
* DATE data type with null indicator
COL9_NI  DS    H
COL9     DS    CL10
* TIME data type with null indicator
COL10_NI DS    H
COL10    DS    CL8
* TIMESTAMP data type with null indicator
COL11_NI DS    H
COL11    DS    CL26
* ROWID data type
COL12_LEN DS   H
```

```
COL12_TXT DS   CL40
         END
```

In PL/I, the copybook looks like this:

```
 DCL 1  EXPORT_EXAMPLE UNALIGNED,
/* INTEGER DATA TYPE WITH NULL INDICATOR                          */
     2 COL1_NULL             BIN(15,0),
     2 COL1                  BIN(31,0),
/* SMALLINT DATA TYPE WITH NULL INDICATOR                         */
     2 COL2_NULL             BIN(15,0),
     2 COL2                  BIN(15,0),
/* FLOATING POINT (4 BYTES) DATA TYPE WITH NULL INDICATOR         */
     2 COL3_NULL             BIN(15,0),
     2 COL3                  FLOAT,
/* FLOATING POINT (8 BYTES) DATA TYPE WITH NULL INDICATOR         */
     2 COL4_NULL             BIN(15,0),
     2 COL4                  FLOAT(8),
/* PACKED DECIMAL DATA TYPE WITH NULL INDICATOR                   */
     2 COL5_NULL             BIN(15,0),
     2 COL5                  FIXED DEC(1,0),
     2 COL5_PAD              CHAR(15),
/* PACKED DECIMAL DATA TYPE WITH NULL INDICATOR                   */
     2 COL6_NULL             BIN(15,0),
     2 COL6                  FIXED DEC(15,7),
     2 COL6_PAD              CHAR(8),
/* CHARACTER DATA TYPE WITH NULL INDICATOR                        */
     2 COL7_NULL             BIN(15,0),
     2 COL7                  CHAR(10),
/* VARCHAR DATA TYPE WITH NULL INDICATOR                          */
     2 COL8_NULL             BIN(15,0),
     2 COL8,
        3 COL8_LEN           BIN(15,0),
        3 COL8_TEXT          CHAR(10),
/* DATE DATA TYPE WITH NULL INDICATOR                             */
     2 COL9_NULL             BIN(15,0),
     2 COL9                  CHAR(10),
/* TIME DATA TYPE WITH NULL INDICATOR                             */
     2 COL10_NULL            BIN(15,0),
     2 COL10                 CHAR(8),
/* TIMESTAMP DATA TYPE WITH NULL INDICATOR                        */
     2 COL11_NULL            BIN(15,0),
     2 COL11                 CHAR(26),
/* ROWID DATA TYPE                                                */
     2 COL12,
        3 COL12_LEN          BIN(15,0),
        3 COL12_TEXT         CHAR(40);
```

The only difference between the first and second set of definitions is the addition of a 2-byte binary field immediately prior to the column definition for those columns that allow Db2® null values.

Related tasks

Copying data to a VSAM or QSAM file on page 279
Using templates with non-Db2 data on page 77

Related references

Export Utility ("From") panel on page 698

How do I export data using a copybook that describes a different data format to that used by the ZDT/Db2 Export Utility?

Scenario: I have data in a Db2® table. I want to export data using a copybook that I have defined, but the copybook describes a different data format to that used by the ZDT/Db2 Export Utility.

You need to code the copybook in either COBOL, HLASM, or PL/I.

When you perform the export with the ZDT/Db2 Export Utility (3.7), enter the name of the data set that contains the copybook in the **To Copybook or Template** section on the Export *"To"* panel. Set the **Template usage** processing option to `1` (Above). If you specify `4` (Generate and save), your copybook will be overwritten. If you have renamed the fields in the copybook, or if data type conversions are required, use template re-mapping to map columns in the source table to fields in the target data set.

If the Db2® data contains one or more nullable columns, you are restricted in the processing for the Db2® null indicator.

If you need the Db2® null indicator in the exported data, you *must* include a 2-byte binary field immediately after the definition for the field in the template that corresponds to the column.

For example, for a CHAR(10) field that allows for the Db2® null value, you would code:

```
* CHARACTER DATA TYPE
     10 COL7                 PIC X(10).
     10 COL7-NULL             PIC S9(4) USAGE COMP.
```

In addition, the 2-byte field must be un-mapped when the export begins. In other words, you must not map a column of the table to this field using template re-mapping; if you do, the ZDT/Db2 Export Utility does not include the null indicator for the column in the exported data set.

If you want the Db2® null indicator mapped to a different position in the export data set, or you want the Db2® null indicator converted into a different data type, you can:

1. Export the data using either the default mapping, or using a user-defined copybook that includes fields defined for the Db2® null indicators as described above.
2. Create a template that describes the final data format.
3. Use the Z Data Tools Base component to copy the data from the export data set into the final format. You may need to use REXX to convert the Db2® null indicators (and possibly other data values) as part of the copy process.

Related tasks

Copying data to a VSAM or QSAM file on page 279
Using templates with non-Db2 data on page 77

Related references

Export Utility ("To") panel on page 701

Chapter 9. Working with lists of Db2® objects

If you want to look at a group of Db2® objects, you can use the Object List utility (3.4) to list or print the objects in which you're interested.

You can use the Object List utility to specify:

- Whether you want the list of objects displayed or printed.
- The type of object you want listed or printed.
- Criteria to restrict the objects listed or printed.
- Options to apply when processing the listed objects.

To display the Object List Utility panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `4` (Object List)

Now that you have the Object List Utility panel displayed, you can start specifying the sorts of Db2® objects you want to display or print.

Note:

1. One of the options available at the time of installing ZDT/Db2 allows for views on Db2® catalog tables to be created for those tables accessed by ZDT/Db2. The set of views includes only those columns within the tables that are needed by ZDT/Db2.

   If this option has been used at installation, it may affect the results obtained when you use the Object List utility. When ZDT/Db2 accesses a view, it shows only the columns of the view, rather than the columns of the table. If there are fewer columns in the view than the underlying table, the information displayed is incomplete.
2. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.

   When you select this option, any display of catalog table information includes every available column, including any columns marked as "unused" or "internal use only" in the *DB2 for z/OS SQL Reference*.

   The default for this option is not selected. Selecting this option overrides any settings for the **Show "unused" catalog table columns** and **Show "internal use only" catalog table columns** options.

Related tasks

Working with object list panels on page 307
Displaying or printing a list of objects on page 304

Related references

Primary Option Menu panel on page 760
Object List Utility panel on page 748
Utility Functions panel on page 915
Editor Options (3 of 8) panel on page 655

Displaying or printing a list of objects

To display or print a list of objects:

1. Specify object identification criteria to identify the objects you want listed:

   Location

   The location of the objects you want listed.

   Leave this field blank if the objects you want listed are at your current location. Otherwise, type the full name of the remote location, an asterisk (*) to display all available remote locations, or a pattern using wildcard characters. If you specify an asterisk on its own or a pattern, ZDT/Db2 displays a selection list showing the locations that match the input specification.

   To select the location you want, type `s` against the required entry and press Enter.

   Owner

   The name of the person who created the Db2® object.

   To select objects regardless of their owner, leave this field blank. Otherwise, to restrict the list of objects by owner, type either the full name of the owner or a pattern using wildcard characters. If this entry field is not blank, ZDT/Db2 limits the objects listed to those matching the owner selection criteria you specify.

   Name

   The name of the Db2® object.

   To select objects regardless of their name, leave this field blank. Otherwise, to restrict the list of objects by name, type either the full name of the object or a pattern using wildcard characters. If this entry field is not blank, ZDT/Db2 limits the objects listed to those matching the name selection criteria you specify.

   Database/collect/schema

   The name of a database, a collection, or a schema. The name can be a full name or a pattern. This name is used to restrict the list of objects to only those objects that are in the specified database, collection, or schema. If you specify a pattern, the list of objects includes any objects that are included in a database or collection or schema whose name matches the pattern specification.

How ZDT/Db2 uses this field depends upon the type of object being listed as specified using the **Object Type** (see step 3 on page 306):

Object type

Field use

Database

Not allowed

Table space

Used as database name

Table, view, or alias

Used as database name

Index

Used as database name

Column

Used as database name

Synonym

Used as database name

Plan

Not allowed

Package

Used as collection name

Collection

Not allowed

DBRM

Not allowed

Schema

Used as schema name

Distinct type

Used as schema name

Function

Used as schema name

Stored procedure

Used as schema name

Trigger

Used as schema name

2. Specify any additional selection criteria to further restrict the objects to be listed by coding a simple SQL SELECT *"where clause"* that specifies a column in the appropriate Db2® system catalog table.

   Column

   Name of column in the relevant Db2® system catalog table. To list the columns in the appropriate Db2® system catalog table, enter an asterisk (*).

   Operator

   A valid SQL operator.

   Value

   The value to be compared with the contents of the column.

   For more information about SQL statements, see the *DB2 for z/OS SQL Reference*.

   For example, to restrict a list of columns to only those columns with a data type of DATE, specify the following values:

```
Additional Selection Criteria:
   Column  . . . . . . . . . COLTYPE___________ Enter * to list catalog columns
   Operator  . . . . . . . . =_________
   Value . . . . . . . . . . 'DATE'_____________________________________________
                             _____________________________________________________
```

3. Nominate the type of object you want listed, and other options to apply to processing of the listed objects.

   Object Type

   Specify the type of Db2® object that you want listed. The numeric indicator for each object type is shown on the Object List Utility panel. This input field is mandatory, but defaults to *"3"* (Table/view/alias).

   Confirm Db2® Object Drop

   Indicates if you want ZDT/Db2 to display a confirmation panel each time you enter a DR(op) command against an object.

   /

   Display a confirmation panel if a DROP command is entered against an object. This is the default setting.

   (blank)

   If a DROP command is entered against an object, issue the command without displaying a confirmation panel.

4. Specify if you want to display or print the list of objects:
   - To display a list of objects, leave the command line blank.
   - To print a list of objects, enter `P` on the command line.

5. Press Enter.

   ZDT/Db2 displays or prints the object list panel for the type of object you have specified.

Related tasks

Using an asterisk (*) or a pattern in entry fields on page 50

Related references

Remote Db2 Location Selection panel on page 791

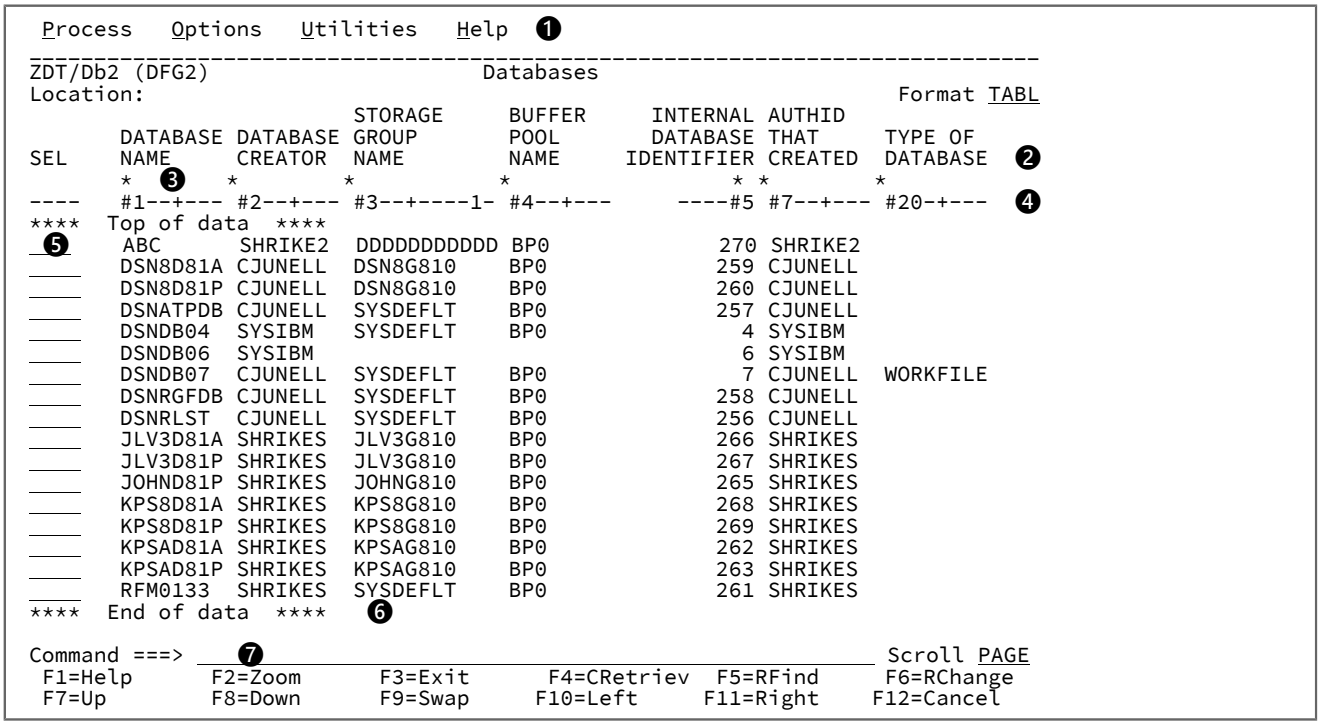
Working with object list panels

After you have completed the necessary steps to display a list of objects, ZDT/Db2 displays the object list panel for the corresponding object type.

The following pages describe the features that are common to all object list panels.

Figure 67: Layout of a typical object list panel on page 307 shows the areas of a typical object list panel.

Figure 67. Layout of a typical object list panel



❶

Action bar. The action bar choices for the object list panels function.

❷

Column heading lines. Shows the name of the column of data shown below. A plus sign ("+"), minus sign ("-"), or both ("+-") indicates that the column is scrollable and may contain more data than is currently displayed.

❸

Filter line. This is where you can specify a filter string for one or more columns to limit the information ZDT/Db2 displays.

❹

Operator line. This is where you can specify an operator for one or more columns to work in conjunction with the corresponding filter pattern.

❺

Line command area.

❻

End of data marker. This marker line indicates the end of the data returned from ZDT/Db2.

❼

Command line. On this line you can specify any Db2® command, ISPF command, or ZDT/Db2 primary command.

Related tasks

Displaying or printing a list of objects on page 304
Working with scrollable display fields on page 43
Limiting the data displayed on page 309
Using the line command area (Cmd) on page 314
Sorting the displayed data on page 323

Related references

Collections panel on page 482
Columns panel on page 503
Databases panel on page 609
Database Request Modules panel on page 607
Distinct Types panel on page 637
Functions panel on page 709
Indexes panel on page 723
Application Packages panel on page 469
Application Plans panel on page 473
Schemas panel on page 812
Stored Procedures panel on page 852
Synonyms panel on page 855

Tables, Views and Aliases panel on page 891
Table Spaces panel on page 879
Triggers panel on page 903

Limiting the data displayed

You can limit the data that ZDT/Db2 displays by *"filtering out"* the data you do not want displayed.

To filter the data, use:

- The *filter line* to specify a filter pattern for one or more columns
-

  And, optionally, the *operator line* to specify an operator to work in conjunction with the corresponding filter pattern.

Using the filter line

On the filter line, the asterisk (*) under a column heading marks an area where you can enter a filter pattern to limit the data displayed. You can specify filter patterns for more than one column if necessary.

For alphanumeric columns, the asterisk is displayed in the leftmost position. For numeric columns, the asterisk is displayed in the rightmost position.

In the filter pattern area, use:

- An asterisk (*) on its own to match all values, or
- A filter pattern using wildcard characters to select only those objects that match the pattern.

You can specify a filter pattern using any of the following special characters:

asterisk (*)

Represents a string of zero or more characters. Acts the same as a percent sign (%).

percent sign (%)

Represents a string of zero or more characters. Acts the same as an asterisk (*).

underscore (_)

Represents any single character, excluding a trailing blank.

When you specify a filter pattern for a column, ZDT/Db2 only selects objects where the contents of that column match the specified filter pattern.

Note: If you have also specified an operator (on the operator line) for the column, ZDT/Db2 uses the operator in conjunction with the filter pattern to match objects (see following section).

For alphanumeric columns, ZDT/Db2 matches the filter pattern you enter with the *leading* characters of the contents of the column for each row. That is, the filter pattern DEF matches 'DEF' and 'DEFG', but not 'CDEF'.

For numeric columns, ZDT/Db2 matches the filter pattern you enter with the *value* of the contents of the column for each row. That is, the search argument 7 matches columns containing 7 but not, for example, 70 or 17.

The following examples show how you can use filter patterns to limit the data ZDT/Db2 displays. (In the examples shown, it is assumed no operator has been specified in the corresponding area on the operator line.)

Pattern filter

Selects objects where column contains…

*****

Any alphanumeric string

ABC*

Any alphanumeric string starting with *"ABC"*

ABC

Any alphanumeric string containing the string *"ABC"*

A__C

Any four-character alphanumeric string where the first character is *"A"* and the last character is *"C"*

%BC_

Any alphanumeric string containing the string *"BC"* followed by one more character

123

For numeric columns: the value 123

23

For numeric columns: the value 23

Using the operator line

You use the operator line to specify an operator that ZDT/Db2 uses in conjunction with the filter pattern specified in the corresponding area on the filter line. You can specify operators for more than one column if necessary. Normally, you only specify an operator for a column where you have also specified a filter pattern.

To enter an operator in the operator line, type the required operator anywhere within the operator area for the required column. Processing occurs left to right. ZDT/Db2 only recognizes the first operator encountered and any other characters entered in the operator field are ignored. For example, if you type *"=>"* in the operator field, ZDT/Db2 treats it as *"="*, and ignores the *">"*.

To remove an operator, overtype the operator with a space.

You can use any of the following operators:

Blank

Means *"like"*. Any *, %, or _ characters in the corresponding filter pattern are treated as special characters. A match occurs if the filter pattern matches the data after considering any special characters.

! (or) ¬

Means *"not like"*. Any *, %, or _ characters in the corresponding filter pattern are treated as special characters. A match occurs if the filter pattern does not match the data after considering any special characters.

=

Means *"equals"*. For numeric data types, special characters in the corresponding filter pattern are not allowed. For non-numeric data types, *, %, and _ characters are allowed but are not treated as special characters. *, %, and _ characters match themselves only.

!=, ¬=, (or) <>

Means *"not equals"*. For numeric data types, special characters in the corresponding filter pattern are not allowed. For non-numeric data types, *, %, and _ characters are allowed but are not treated as special characters. *, %, and _ characters match themselves only.

<

Means *"less than"*. For numeric data types, special characters in the corresponding filter pattern are not allowed. For non-numeric data types, *, %, and _ characters are allowed but are not treated as special characters. *, %, and _ characters match themselves only.

<=, !>, (or) ¬>

Means *"less than or equal to"*. For numeric data types, special characters in the corresponding filter pattern are not allowed. For non-numeric data types, *, %, and _ characters are allowed but are not treated as special characters. *, %, and _ characters match themselves only.

>=, !<, (or) ¬<

Means *"greater than or equal to"*. For numeric data types, special characters in the corresponding filter pattern are not allowed. For non-numeric data types, *, %, and _ characters are allowed but are not treated as special characters. *, %, and _ characters match themselves only.

>

Means *"greater than"*. For numeric data types, special characters in the corresponding filter pattern are not allowed. For non-numeric data types, *, %, and _ characters are allowed but are not treated as special characters. *, %, and _ characters match themselves only.

You can also enter the following commands in the operator field for a column:

X

Excludes the column from the data display. The effect is the same as editing the template and de-selecting the column, followed by a re-display of the data.

W*

Resets the display width for the column to the maximum value for the column. The command is only effective when entered against a column that has a non-numeric data type.

W*nnn*

Sets the display width for the column to *nnn* (*nnn* must be an integer between 6 and the maximum possible display width for the column). The command is only effective when entered against a column that has a non-numeric data type.

When you specify an operator for a column, ZDT/Db2 first applies the operator to the corresponding filter pattern and then only selects objects where the contents of that column match the resultant filter pattern.

The following examples show how you can use operators, in conjunction with filter patterns, to limit the data ZDT/Db2 displays.

Pattern filter (operator)

Selects objects where column…

ABC* (blank)

Contains any alphanumeric string starting with *"ABC"*

ABC* (=)

Contains the string *"ABC**"*

ABC* (¬)

Does not contain the string *"ABC**"*

A (>)

Contains an alphanumeric string whose first character is greater than *"A"*

A_C (¬>)

Contains an alphanumeric string whose first three characters are less than or equal to the string *"A_C"*

123 (=)

For numeric columns: contains the value 123

23 (<)

For numeric columns: contains a value less than 23

Let's take an example. Figure 67: Layout of a typical object list panel on page 307 shows a list of database object types. To limit the columns displayed to those where:

- The database creator is SHRIKES
- The storage group name starts with the letter K
- The internal database identifier is greater than 264

Enter the following filter patterns and operators:

1. Under the column heading DATABASE CREATOR, type `SHRIKES` on the filter line.
2. Under the column heading STORAGE GROUP NAME, type `K*` on the filter line.
3. Under the column heading INTERNAL DATABASE IDENTIFIER, type `264` on the filter line and type `>` on the operator line.

Figure 68. Object list: databases

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Databases
Location:                                                      Format TABL
                        STORAGE     BUFFER     INTERNAL AUTHID
       DATABASE DATABASE GROUP       POOL       DATABASE THAT     TYPE OF
SEL    NAME     CREATOR  NAME        NAME     IDENTIFIER CREATED  DATABASE
       *        SHRIKES  K*          *             264 *        *
----   #1--+--- #2--+--- #3--+----1- #4--+---     ---- > #7--+--- #20-+---
****  Top of data  ****
_____  ABC      SHRIKE2  DDDDDDDDDDD BP0             270 SHRIKE2
_____  DSN8D81A CJUNELL  DSN8G810    BP0             259 CJUNELL
_____  DSN8D81P CJUNELL  DSN8G810    BP0             260 CJUNELL
_____  DSNATPDB CJUNELL  SYSDEFLT    BP0             257 CJUNELL
_____  DSNDB04  SYSIBM   SYSDEFLT    BP0               4 SYSIBM
_____  DSNDB06  SYSIBM                                 6 SYSIBM
_____  DSNDB07  CJUNELL  SYSDEFLT    BP0               7 CJUNELL  WORKFILE
_____  DSNRGFDB CJUNELL  SYSDEFLT    BP0             258 CJUNELL
_____  DSNRLST  CJUNELL  SYSDEFLT    BP0             256 CJUNELL
_____  JLV3D81A SHRIKES  JLV3G810    BP0             266 SHRIKES
_____  JLV3D81P SHRIKES  JLV3G810    BP0             267 SHRIKES
_____  JOHND81P SHRIKES  JOHNG810    BP0             265 SHRIKES
_____  KPS8D81A SHRIKES  KPS8G810    BP0             268 SHRIKES
_____  KPS8D81P SHRIKES  KPS8G810    BP0             269 SHRIKES
_____  KPSAD81A SHRIKES  KPSAG810    BP0             262 SHRIKES
_____  KPSAD81P SHRIKES  KPSAG810    BP0             263 SHRIKES
_____  RFM0133  SHRIKES  SYSDEFLT    BP0             261 SHRIKES
****  End of data  ****

Command ===> _____________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

4. Press Enter. A list of database object types is displayed, restricted to the search criteria you specified:

   Figure 69. Object List: databases (list restricted using filter patterns and operators)

```
 Process   Options   Utilities   Help
_______________________________________________________________________________
ZDT/Db2 (DFG2)                    Databases
Location:                                                        Format TABL
                          STORAGE     BUFFER     INTERNAL AUTHID
       DATABASE DATABASE  GROUP       POOL       DATABASE THAT     TYPE OF
SEL    NAME     CREATOR   NAME        NAME     IDENTIFIER CREATED  DATABASE
       *        SHRIKES   K*          *             264 *        *
----   #1--+--- #2--+---  #3--+----1- #4--+---      >---#5 #7--+--- #20-+---
****  Top of data  ****
_____  KPS8D81A SHRIKES  KPS8G810    BP0              268 SHRIKES
_____  KPS8D81P SHRIKES  KPS8G810    BP0              269 SHRIKES
****  End of data  ****









Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Locating a row or column in a list of Db2® objects

When you display data with the Object List utility, you can use the LOCATE primary command to locate either a row or a column in a similar way to when you are viewing data with ZDT/Db2 View or Edit.

Related references

Locating a row on page 142
Locating a column on page 142
LOCATE primary command on page 978

Finding a string in a list of Db2® objects

When you display data with the Object List utility, you can use the FIND primary command to find a specific string in a similar way to when you are viewing data with ZDT/Db2 View or Edit.

Related references

Finding a string on page 146
FIND primary command on page 953

Using the line command area (Cmd)

In the line command area (under the heading **Cmd**) for a listed object, you can enter a line command or one of the general line commands (?, =, or ROW).

Line command

Table 22: Object list line commands on page 315 shows the line commands you can use with each type of object.

Table 22. Object list line commands

Command	Description	Database	Table space	Table/View	Index	Column	Synonym	Plan	Package	Collection	DBRM	Schema	Distinct Type	Function	Procedure	Trigger	Storage Group	Sequence
A	Alter an object.	Y	Y	Y[15]	Y									Y	Y			
ADD	Add table check constraint.			Y[16]														
ALS	Show aliases of an object.			Y														
AT	Show auxiliary tables for LOB columns.			Y[15]		Y												
B	Browse the object.			Y			Y[17]				Y							
BI	Bind the object.							Y	Y									
BT	Show base table for auxiliary table.			Y[15]														
C	Copy table.			Y			Y[17]											
CBI	Copy and bind the object.								Y									
CDI	Show column distribution statistics (as generated by RUNSTATS).			Y														
CH	Show child (dependent) tables.			Y														
CHR	Show child relationships (referential constraints).			Y														
CK	Show information about table check constraints.			Y														
COL	Show the columns for the object.			Y	Y	Y[18]												
COM	Comment on an object.			Y		Y							Y	Y	Y	Y		
CR	Create an object like this object.	Y	Y	Y[15]	Y		Y						Y	Y	Y	Y	Y	Y

15. Not available for Materialized Query Table object.

16. Only available from a list of table constraints.

17. The synonym name is copied to the function panel if the location is unspecified and the owner is the SQLID; otherwise, the synonym name is resolved and the resolved name is copied.

18. Only available from a list of object privileges.

Table 22. Object list line commands

(continued)

Command	Description	Database	Table space	Table/View	Index	Column	Synonym	Plan	Package	Collection	DBRM	Schema	Distinct Type	Function	Procedure	Trigger	Storage Group	Sequence
CRA	Create an alias for an object.			Y			Y											
CRS	Create a table space in the database.	Y																
CRT	Create a table in the table space.		Y															
CRX	Create an index.			Y														
CS	Describe the columns for an object (abbreviation of information provided by COL command).			Y														
D	Show the database for an object.	Y[18]	Y	Y	Y												Y	
DEP	Show the dependencies on an object.			Y[16]				Y	Y									Y
DI	Display distribution statistics of a column.					Y												
DR	Drop the object/constraint.	Y	Y	Y	Y		Y						Y	Y	Y	Y	Y	Y
DS	Display the database structure.	Y	Y															
DT	Show the source data type (columns) or distinct types (schemas).					Y						Y						
E	Edit the object.			Y			Y[17]											
EDC	Show enabled/disabled connections.							Y	Y									
F	Show functions.											Y						
FC	Show the foreign key From Column.			Y[19]														

19. Only available from a list of foreign keys.

Table 22. Object list line commands

(continued)

Command	Description	Database	Table space	Table/View	Index	Column	Synonym	Plan	Package	Collection	DBRM	Schema	Distinct Type	Function	Procedure	Trigger	Storage Group	Sequence
FK	Show information about the referential integrity defined for foreign keys.			Y														
FRE	Free the object.							Y	Y									
G	Grant privileges for the object.	Y	Y	Y		Y		Y	Y	Y			Y	Y	Y		Y	Y
GEN	Generate SQL for object from Db2® catalog.	Y	Y	Y[15]	Y							Y	Y	Y	Y			
H	Show the homonyms (like-named objects) for the object.					Y												
I	Show detailed information about the object.	Y	Y	Y	Y	Y	Y	Y	Y		Y		Y	Y	Y	Y		
ICS	Show the status of image copies for the object.	Y	Y		Y													
L	Show the collection for the object or, for tables, show the rows of this table.									Y[18]								
LAB	Label the object.			Y		Y												
LC	Show LISTCAT information for the data set.	Y[20]	Y[20]															
LPT	List the PLAN_TABLE table for the object.							Y	Y									
M	Show the DBRMs for the object.							Y										
O	Show related stored procedures.											Y						
P	Display information about the privileges for this object.	Y	Y	Y		Y		Y	Y	Y		Y	Y	Y	Y		Y	

20. Only available from a list of image copies.

Table 22. Object list line commands

(continued)

Command	Description	Database	Table space	Table/View	Index	Column	Synonym	Plan	Package	Collection	DBRM	Schema	Distinct Type	Function	Procedure	Trigger	Storage Group	Sequence
PA	Show parent tables.			Y														
PAR	Show parent relations (referential constraints).			Y														
PARM	Show the routine parameters.													Y	Y			
PC	Show the column privileges for this object.			Y[18]														
PK	Show the primary key for this table.			Y														
PKG	Show the packages for the object.			Y	Y			Y	Y[18]	Y				Y	Y	Y		
PKL	Show the package lists for the object.							Y	Y	Y								
PL	Show the plans for the object.			Y	Y			Y[18]	Y	Y	Y							
PR	Print table.			Y			Y[17]											
PS	Display information about the privileges for the schema for this object.												Y	Y	Y	Y		
PST	Show partition statistics.			Y		Y												
PT	Show the parts of the object.		Y		Y												Y	
R	Revoke the privilege for this object.[18]	Y	Y	Y				Y	Y	Y		Y	Y	Y	Y			
RBI	Rebind the object.							Y	Y									
RDT	Return data type.													Y				
ROW (or "/")	Show all the columns in this row of the result table.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
RT	Recover TOCOPY.	Y[20]	Y[20]															

Table 22. Object list line commands

(continued)

Command	Description	Database	Table space	Table/View	Index	Column	Synonym	Plan	Package	Collection	DBRM	Schema	Distinct Type	Function	Procedure	Trigger	Storage Group	Sequence
S	Show the table spaces for the object.	Y	Y[18]	Y				Y	Y								Y	
SCH	Show the schema the object.											Y[18]	Y	Y	Y[18]	Y		
SEL	Build SQL SELECT statement for this object.			Y			Y[17]											
SEQ	Show identity column information.					Y												
SQL	Show the SQL statements in the object.							Y	Y	Y	Y					Y		
SYN	Show the synonyms for the object.			Y				Y	Y									
T	Show the tables for the object.	Y	Y	Y[21]	Y	Y	Y	Y	Y				Y			Y		
TC	Show the foreign key To Column.			Y[19]														
TR	Show triggers.			Y[15]								Y						
UTL	Run a Db2® utility job against the object.		Y	Y	Y		Y[17]											
V	View the object.			Y			Y[17]											
VER	Show the versions of the object.								Y									
VOL	Show volumes.															Y	Y	
VS	Show how the view was created.			Y[22]														
VW	Show views.			Y			Y											
X	Show the indexes for the object.	Y		Y		Y		Y	Y								Y	

21. Only available when the object is a view or an alias, or from a list of object privileges.

22. Only available when the object is a view.

Table 22. Object list line commands

(continued)

Command	Description	Database	Table space	Table/View	Index	Column	Synonym	Plan	Package	Collection	DBRM	Schema	Distinct Type	Function	Procedure	Trigger	Storage Group	Sequence
XC	Show the indexes and index columns for the object.			Y														
XPT	Show index parts.															Y	Y	

Related tasks

Line command query (?) on page 320
Equal (=) on page 321
ROW on page 322

Line command query (?)

If you enter ? in the line command area, ZDT/Db2 displays a list of the line commands you can enter for this object type. Figure 70: Using the ? line command to display a list of valid line commands on page 320 shows an example of a list of valid line commands (in this case, for a synonym object type).

Figure 70. Using the ? line command to display a list of valid line commands

```
 Process   Options   Utilities   Help
_______________________________________________________________________
ZDT/Db2 (DFG2)            Object List Line Commands           Row 1 to 6 of 6

Type S against the required line command and press Enter.

Sel Command  Description

_   B        Browse table
_   C        Copy table
_   CR       Create a synonym
_   CRA      Create an alias
_   DR       Drop synonym
_   E        Edit table
_   I        Details about synonym
_   PR       Print table
_   ROW      Show all columns for this row
_   SEL      Build a SELECT statement
_   T        Show tables
_   UTL      Utility functions

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

To select a command from the list of valid commands, enter `S` in the **Sel** column.

Related references

Object List Line Commands panel on page 742
Table 22: Object list line commands on page 315

Equal (=)

If you enter = in the line command area, ZDT/Db2 runs the previous line command.

Figure 71: Using the equal (=) line command on page 321 shows the use of the = line command. The previous line command that ZDT/Db2 ran was DS (to display the structure of the data base DSNRLST). The asterisk is displayed in the first position of the line command area after ZDT/Db2 has run the command and you have returned to the original panel. If you enter = in the line command area for another listed object (in this case, the data base FMN2), ZDT/Db2 runs the DS command for that object.

Figure 71. Using the equal (=) line command

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                    Databases                  Row 1 to 10 of 63

Location:

                                                                      Index
                         Storage  Buffer          Created             Buffer
Cmd   Name     Owner   + Group    Pool       DBID By      + T E Pool
      *_______ *_______ *_______ *_______ _____* *_______ * * *_______
_____ DSNDB04  SYSIBM    SYSDEFLT BP0            4 SYSIBM       BP0
_____ DSNDB06  SYSIBM                            6 SYSIBM     E BP0
_____ DSNDB07  HFMUSER   SYSDEFLT BP0            7 HFMUSER  W   BP0
*S___ DSNRLST  HFMUSER   SYSDEFLT BP0          256 HFMUSER    E BP0
_____ DSNRGFDB HFMUSER   SYSDEFLT BP0          257 HFMUSER    E BP0
=____ FMN2     HFMUSER   SG01     BP0          258 HFMUSER    E BP0
_____ FMDB2    HFMUSER   SYSDEFLT BP0          259 HFMUSER    E BP0
_____ DSN8D61P BRADB     DSN8G610 BP0          261 BRADB      E BP0
_____ DSN8D61A BRADB     DSN8G610 BP0          260 BRADB      E BP0
_____ DSN8D61L BRADB     DSN8G610 BP0          262 BRADB      E BP0

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

If you enter the "=" line command multiple times, as shown in Figure 72: Using the equal (=) line command multiple times on page 322, ZDT/Db2 runs the line command (in this case, DS) consecutively for each of the nominated objects in the list without returning to the original panel between each execution.

Figure 72. Using the equal (=) line command multiple times

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                    Databases                  Row 1 to 10 of 63

Location:

                                                                      Index
                         Storage  Buffer          Created             Buffer
Cmd   Name     Owner  + Group    Pool       DBID By     + T E Pool
      *_______ *_______ *_______ *_______ _____* *_______ * * *_______
_____ DSNDB04  SYSIBM   SYSDEFLT BP0            4 SYSIBM              BP0
_____ DSNDB06  SYSIBM                           6 SYSIBM     E BP0
_____ DSNDB07  HFMUSER  SYSDEFLT BP0            7 HFMUSER  W   BP0
*S___ DSNRLST  HFMUSER  SYSDEFLT BP0          256 HFMUSER    E BP0
_____ DSNRGFDB HFMUSER  SYSDEFLT BP0          257 HFMUSER    E BP0
=____ FMN2     HFMUSER  SG01     BP0          258 HFMUSER    E BP0
_____ FMDB2    HFMUSER  SYSDEFLT BP0          259 HFMUSER    E BP0
=____ DSN8D61P BRADB    DSN8G610 BP0          261 BRADB      E BP0
=____ DSN8D61A BRADB    DSN8G610 BP0          260 BRADB      E BP0
_____ DSN8D61L BRADB    DSN8G610 BP0          262 BRADB      E BP0

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

ROW

If you enter `ROW` in the line command area, ZDT/Db2 displays the column name (and its value) as held in the Db2® catalog table for each column of the specified object,

You can enter more than one ROW line command at a time.

Figure 73: Using the ROW line command on page 322 shows the use of the ROW line command, placed in the line command area for the synonym object TCONA.

Figure 73. Using the ROW line command

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                     Synonyms                  Row 1 to 11 of 36

Location:

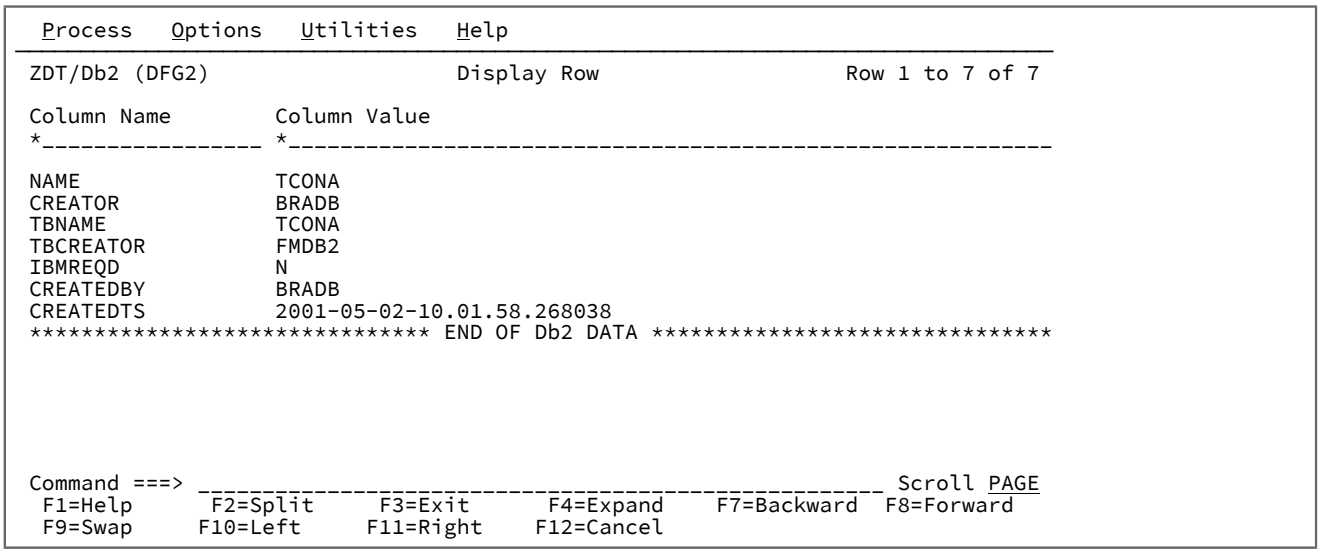
                                                                      Created
Cmd   Synonym          + Owner  + Table/View Name  + Owner  +       By     +
      *________________ *_______ *________________ *_______       *_______

_____ DEPT               HARFOUJ  DEPT               DSN8810       HARFOUJ
_____ DEPT               KIEFERA  DEPT               DSN8810       KIEFERA
_____ SDEPT              KIEFERA  DEPT               DSN8810       KIEFERA
_____ VPSTRDE1           BRADB    VPSTRDE1           FMDB2         BRADB
_____ VPSTRDE2           BRADB    VPSTRDE2           FMDB2         BRADB
_____ VASTRDE1           BRADB    VASTRDE1           FMDB2         BRADB
_____ EMP                BRADB    EMP                FMDB2         BRADB
_____ VPHONE             BRADB    VPHONE             FMDB2         BRADB
ROW__ TCONA              BRADB    TCONA              FMDB2         BRADB
_____ DEPT               BRADB    DEPT               FMDB2         BRADB
_____ VDEPT              BRADB    VDEPT              FMDB2         BRADB

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 74: Result of using the ROW line command on page 323 shows the result. In this case, the column names are from the Db2® catalog table, SYSSYNONYMS.

Figure 74. Result of using the ROW line command



You can limit the number of entries shown on the panel by entering search criteria in the areas marked by an asterisk (*) immediately below the headings.

Related tasks

Limiting the data displayed on page 309

Sorting the displayed data

Use the SORT primary command to sort a column in the current object list.

To sort one column (in ascending sequence):

1. Type `SORT` on the command line.
2. Place the cursor within the column you want to sort.
3. Press Enter.

   ZDT/Db2 sorts the column in ascending sequence.

To sort multiple columns, in ascending or descending sequence:

1. Type `SORT` on the command line.
2. Ensure the cursor is not positioned within any column.

3. Press Enter:

   ZDT/Db2 displays the Sort Fields panel.

   You can limit the number of entries shown on the panel by entering search criteria in the areas marked by an asterisk (*) immediately below the headings.

Related tasks

Limiting the data displayed on page 309

Related references

Sort Fields panel on page 841
SORT primary command on page 1003

Reverse engineering

Reverse engineering lets you reverse engineer the Db2® objects in your database catalog.

Reverse engineering generates the SQL statements necessary to re-create a Db2® object.

Use the GEN line command on the corresponding object list panel

to reverse engineer any of the following objects:

- Database
- Table space
- Table or view
- Index
- Schema
- Distinct type

- Function
- Stored procedure

Note:
The GEN line command applies to objects at the current server only. If you enter the GEN line command against an object at a remote location, the location information is discarded when the Generate SQL From Db2® Catalog panel is displayed.

To generate DDL for objects at a remote location, first connect to that location, then use the GEN line command.

Figure 75: GEN line command entered on Databases object list panel on page 325 shows the GEN line command entered on a database object list panel:

Figure 75. GEN line command entered on Databases object list panel

```
 Process   Options   Utilities   Help
_______________________________________________________________________________
ZDT/Db2 (DFG2)                  Databases                Row 51 to 54 of 54

Location:

                                                           Index
                         Storage  Buffer          Created  Buffer
Cmd   Name     Owner    Group    Pool       DBID By     + T E Pool
      *_______ *______+ *_______ *_______ _____* *_______ * * *_______
_____ CONVDB4  OSPEED   SYSDEFLT BP0          302 OSPEED     E BP0
GEN__ JOHNLEDP SIROED   JOHNLEG0 BP0          304 SIROED     E BP0
_____ GPCOPYL  GRAHAMP  GRAHAMP0 BP0          276 GRAHAMP    E BP0
_____ UTILBIG1 OSPEED   DB2BIG   BP0          306 OSPEED     E BP0
****************************** END OF Db2 DATA *******************************




Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Typical uses for reverse engineering include:

- Extracting the DDL for an object before changes are made, so that the changes are applied to the current definition and are available for fallback purposes.
- Move Db2® objects to another Db2® subsystem. By using reverse engineering (together with the Db2® table unload and load facilities), objects can be moved after a few manual modifications to the generated SQL and batch jobs.

When extracting database, table space, and table objects, all dependent objects can also be generated; this includes table spaces, tables, indexes, views, synonyms, aliases, referential constraints, table checks, and table triggers. When extracting objects in schemas, reverse engineering can extract the dependent distinct types, functions, and stored procedures. All authorizations to these objects can also be generated.

You can generate the SQL statements using a batch or online job.

If you are using ZDT/Db2 to extract several objects from a large catalog, batch jobs are recommended.

If you use the GEN line command, ZDT/Db2 displays the Generate SQL From Db2® Catalog panel.

On the Generate SQL From Db2® Catalog panel, you can:

- As an option, specify new values for the :
  - Storage group (possibly using a different storage group for table spaces and index spaces).
  - Database.
  - Specify a new object owner. If specified, the new owner is used whenever an object is created.
  - Specify a new schema name (where applicable). If specified, the new schema is used whenever an object is created.
  - Specify the data set in which ZDT/Db2 places the generated SQL.
  - Specify whether the SQL generation runs as a batch or online job.
  - Specify how often reverse engineering adds an SQL COMMIT statement to the generated SQL.
  - Specify whether Db2® default parameters are removed or kept in the generated SQL.

If you specify an execution mode of BATCH, ZDT/Db2 generates a batch job

and displays the job in an ISPF Edit session, ready for any modifications you need to make before submitting the job for execution. If you specify TSO, ZDT/Db2 generates the SQL statements online and displays the results.

Related tasks

Performance tips on page 326

Related references

Generate SQL From Db2 Catalog panel on page 712

Performance tips

To improve performance of the extract program, consider adding the following indexes to the Db2® catalog:

On SYSDBAUTH

(database(,grantor,grantee))

On SYSCHECKS

(tbowner,tbname)

On SYSRELS

(creator,tbname,relname)

On SYSRESAUTH

(qualifier,name(,grantor,grantee,obtype))

On SYSTABAUTH

(tcreator,ttname(,grantor,grantee,granteetype))

On SYSCOLAUTH

(creator,tname(,dategranted,timegranted))

The recommended mode of operation is batch, even if only a few objects are requested. This is because the design process is based on the standard Db2® catalog indexes, which means that some parts of the process scan the catalog tables instead of doing a direct reference.

Considerations

The following considerations apply to reverse engineering:

- ZDT/Db2 does not extract IDCAMS DEFINE CLUSTER statements for VCAT-defined table spaces and indexes.
- The generated SQL for table spaces and indexes defined with a DSETPASS (password) contains a SPUFI comment line such as:

  ```
  --     DSETPASS XXXXXXXX
  ```

  ZDT/Db2 does not reveal the data set password in the catalog; ZDT/Db2 generates the comment line and issues a warning.
- The ability to generate actually allocated or actually used space allocations depends on information in the Db2® catalog. The actual data set sizes for table spaces or index spaces are not retrieved. Only use these options if you have recently run STOSPACE and RUNSTATS for the selected objects.

Chapter 10. Printing

ZDT/Db2 provides various utilities that allow you to print or view printed output:

- To print the contents of a Db2® table or view, use the Print utility function (3.1).
- To browse, and optionally purge, the print output from your current ZDT/Db2 editor session held in the print data set, use the Print Browse utility function (3.11).
- To print an audit trail report using the contents of the audit trail data set, use the Audit trail utility function (3.10).

Each of these utilities is described in the remainder of this chapter.

Related tasks

Printing the contents of a Db2 table on page 328
Looking at the print output from your ZDT/Db2 session on page 333
Printing an audit trail report on page 335

Printing the contents of a Db2® table

To print the contents of a Db2® table or view, use the Print utility function (3.1).

When you print data, you can:

- Select which rows to print.
- Select which columns to print and the order in which they are arranged on the listing.
- Print Db2® tables or views with columns formatted according to column data type.
- Provide customized column headings.
- Limit the number of rows printed:

  Limit the print

  You can limit the number of rows that are printed by specifying the number of rows to print.

  To print all the rows of the Db2® object, specify `0` (zero) or `ALL`.

  Select rows

  You can select the rows you want to print by specifying *row selection criteria*.

To display the Print Utility panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `1` (Print)

Now that you have the Print Utility panel displayed, you can specify the name of the Db2® table or view you want to print, how many rows you want to print, and other options to control how the data is printed.

When you have finished specifying these details, press Enter.

If you selected **Edit options** (either by using a "/", or by having previously used the *"A"* selection character for this option), before printing the data, the Print Utility Options panel is displayed.

If you selected **Edit template** (either by using a "/", or by having previously used the *"A"* selection character for this option), before printing the data, the Column Selection/Edit panel is displayed.

If you selected **Batch execution** (either by using a "/", or by having previously used the *"A"* selection character for this option), the generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

Otherwise (if you selected none of the preceding options), ZDT/Db2 prints the data.

Formatting the print output

Formatting of the print output is controlled by the following:

Settings: Print settings (option 0.0.1)

Several processing options, that you can specify using the Set Print Processing Options panel, affect the printed output:

Output destination

Where you want the printed output sent:

SYSPRINT

Send print output to the current SYSPRINT allocation.

Typically, SYSPRINT is allocated to the terminal, making this option synonymous with TERMINAL. However, you can allocate SYSPRINT in other ways.

Terminal

Send print output to the terminal.

Data set

Output is accumulated in the print data set specified in the **Data set name** field. This data set can browsed using the PB command or sent to the JES spool queue for printing by issuing the PRINT command while browsing the data set. (The output is sent to the JES spool queue class specified in the Output class field.)

REXX

Send print output to the REXX stem variable FILEM.

Choose this option when you want to control your printing from within a REXX program.

Record length

How many columns wide the output is

Page skip

Whether output from each function starts on a new page

Wide print

Whether the maximum print line length for print output is used

Translate non-printable chars

Whether non-printable characters are translated to blanks

Uppercase message text

Whether all message text is translated to uppercase.

Data set DISP

Whether the print output is appended to the existing data set (MOD) or replaces it (OLD).

Note: This option only affects output sent to the print data set specified in the PRINTDSN option.

Dump format

Which format (updown or across) is used for hexadecimal print output

Data set name

The print data set where print output is directed when the PRINTOUT print option is set to SYSOUT=*c*

Output class

The class of the JES spool queue to be used when the PRINT command is issued while browsing the temporary print data set.

Lines per page

How many lines per page the output has

Record limits

Which part of each record to print

You can view the print data set using Print Browse (option 3.11). While you are using the Print Browse, you can use the PURGE primary command to clear the print data set. To transfer the contents of the print data set to a SYSOUT class, press the Print function key (F4).

Print Mode

This entry field on the Print Utility panel determines the format of the printed output.

Related references

Set Print Processing Options panel on page 827
PURGE primary command on page 990

Editing the template

Before you print the data, you can edit the template to specify what data is printed and how it is displayed. For example, you might only want to print certain columns, specify different headings for them, and change the sequence in which the columns are printed.

To edit the template, select the **Edit template** option on the Print Utility panel. The Column Selection/Edit panel is displayed.

Running Print in batch

If you want to print large Db2® tables, you can perform the print in a batch job.

If you select the **Batch execution** option on the Print Utility panel, the Print utility builds the necessary JCL to perform the print function in a batch job. The generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing.

Related tasks

Editing the template for the "From" table on page 248
Specifying a Db2 object name on page 46
Specifying a data set and a member name on page 52
Print settings (option 0.1) on page 64
Working with templates on page 74
Looking at the print output from your ZDT/Db2 session on page 333

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Print Utility panel on page 764
Column Selection/Edit panel on page 497

Printing rows from a Db2® table

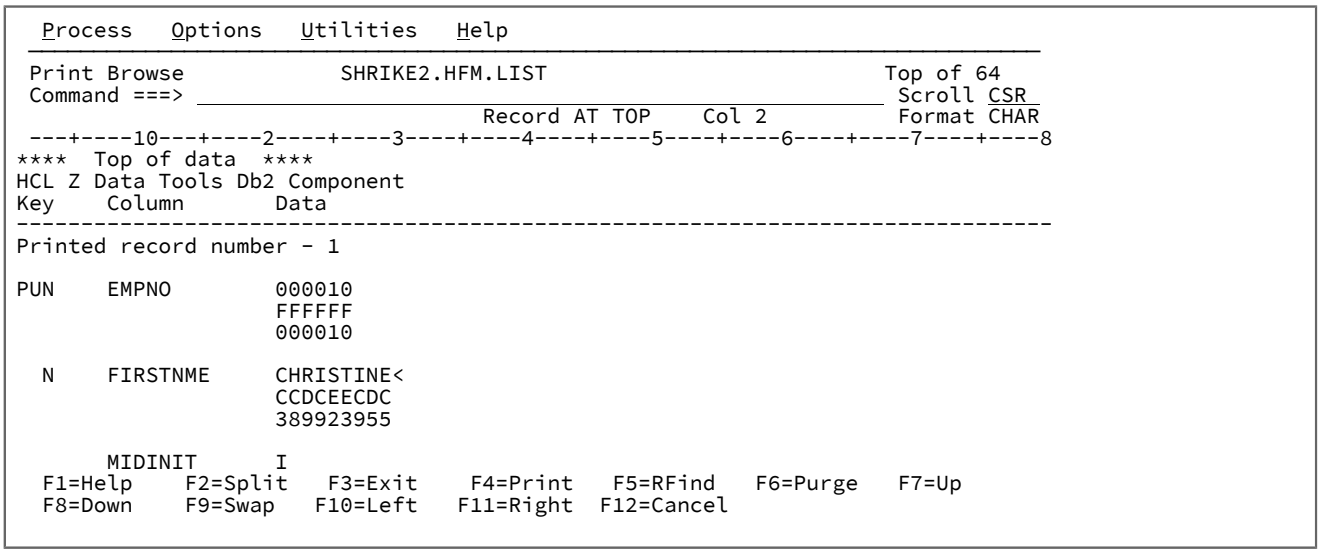
To print rows from a Db2® table, use:

- The RD (*"Record Dump"*) primary command to print the data in dump format
- The RP (*"Record Print"*) primary command to print the data in character format

Examples

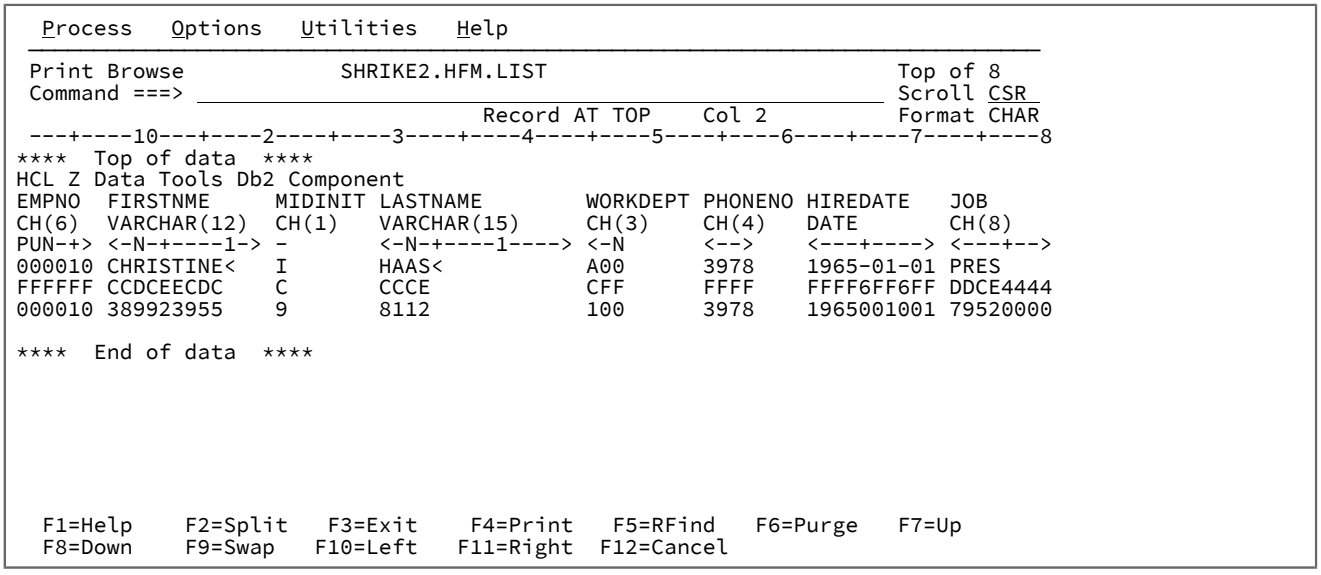
Here is an example of RD output when the display format is SNGL:

Figure 76. Sample printed output from RD primary command (SNGL display format)



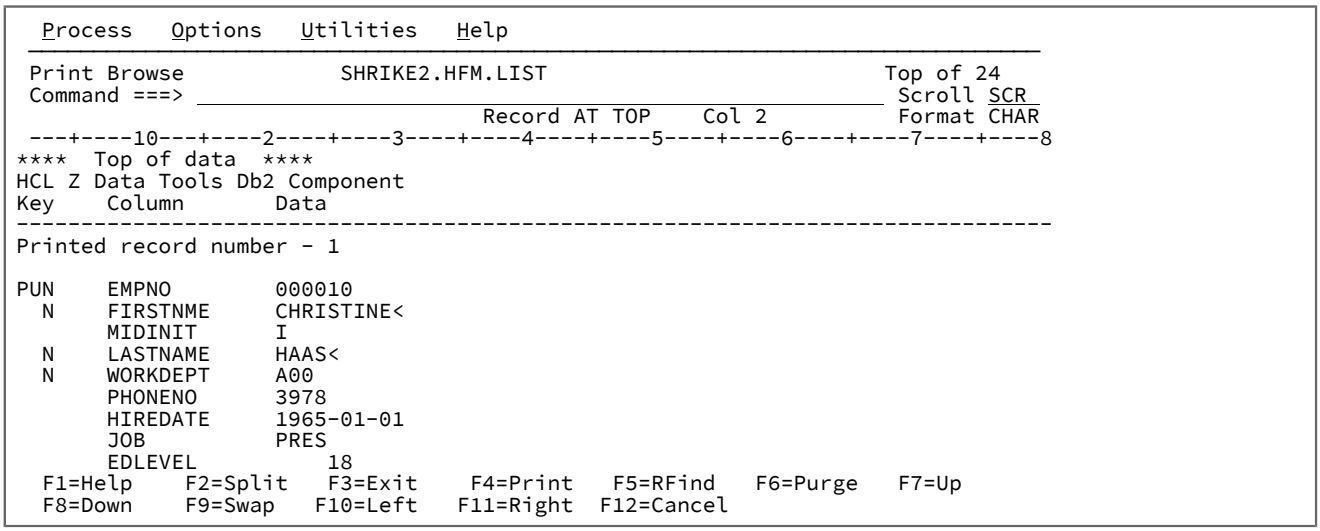
Here is an example of RD output when the display format is TABL:

Figure 77. Sample printed output from RD primary command (TABL display format)



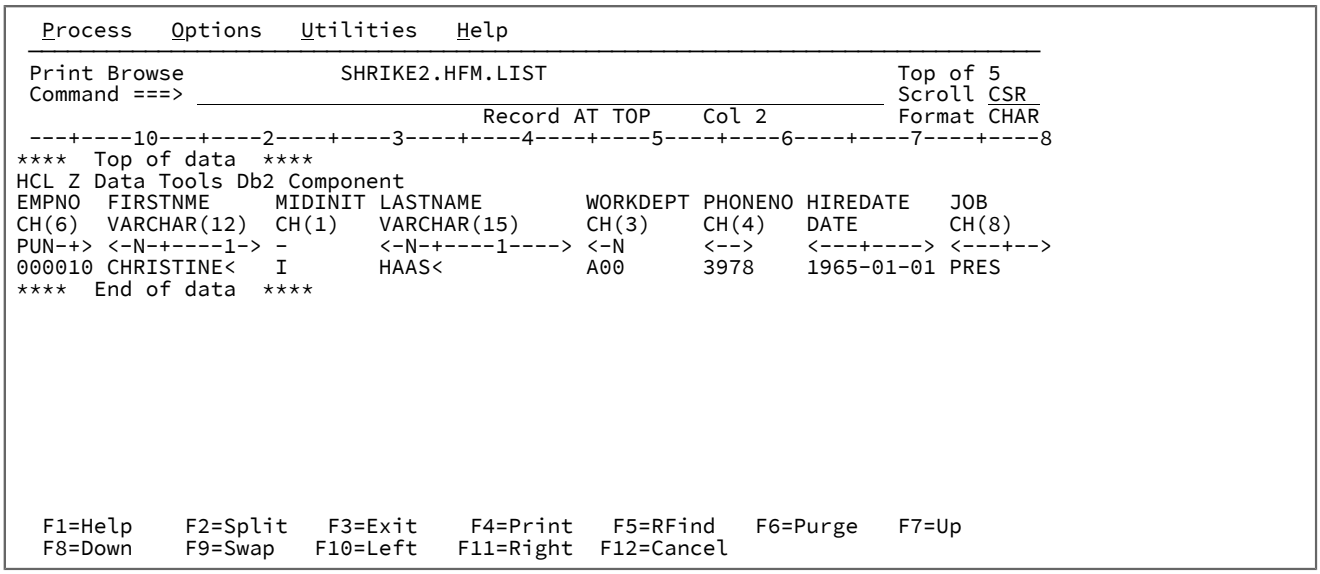
Here is an example of RP output when the display format is SNGL:

Figure 78. Sample printed output from RP primary command (SNGL display format)



Here is an example of RP output when the display format is TABL:

Figure 79. Sample printed output from RP primary command (TABL display format)



Related references

RD primary command on page 991
RP primary command on page 997

Looking at the print output from your ZDT/Db2 session

To browse, and optionally purge, the print output from your current ZDT/Db2 editor session held in the print data set:

- Issue the PB primary command, or
- Use the Print Browse utility function (3.11)

You specify the print data set using the **PRINTDSN** entry field on the Set Print Processing Options panel.

If you use the PB primary command or the Print Browse utility when you have not created any print output, ZDT/Db2 displays the message `Empty print data set`.

The following primary commands are available with Print Browse:

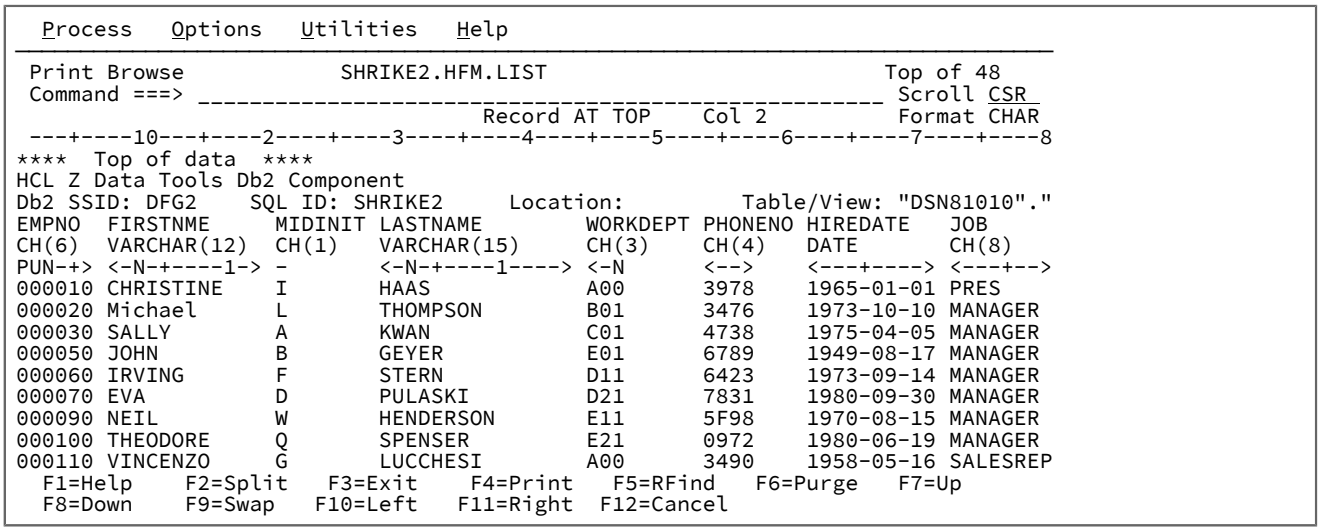
- FIND
- LOCATE
- PRINT
- PURGE

To select the Print Browse utility:

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `11` (Print browse)

Figure 80: Print Browse: sample output on page 334 shows an example of some printed output displayed by Print Browse.

Figure 80. Print Browse: sample output



Related tasks

Printing the contents of a Db2 table on page 328
Printing an audit trail report on page 335

Related references

Primary Option Menu panel on page 760
Utility Functions panel on page 915
Set Print Processing Options panel on page 827
FIND primary command on page 953

LOCATE primary command on page 978
PRINT primary command on page 989
PURGE primary command on page 990

Printing an audit trail report

To print an audit trail report using the contents of the audit trail data set, use the Audit trail utility function (3.10).

An audit trail data set is created if the **Create an audit trail** option is selected when you use:

- View (option 1)
- Edit (option 2)
- Copy (option 3.3)



Note: You can create an audit trail data set yourself using the batch job HFMSMFX (supplied with ZDT/Db2). This job creates an audit trail data set using data from the SMF log file. For details, see the *Z Data Tools Customization Guide*.

To display the Print Audit Trail panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `10` (Audit trail)

To print an audit trail report:

1. Specify the name of the audit trail data set that contains the audit trail data you want to print (**Data set name**).

   If ZDT/Db2 created the audit trail data set (as a result of ZDT/Db2 view, edit, or copy activity), the name is in the format

   prefix`.HFM2AUD.`*ssid*`.D`*yymmdd*`.T`*hhmmss*

   or

   userid`.HFM2AUD.`*ssid*`.D`*yymmdd*`.T`*hhmmss*

   where:

   prefix

   is the TSO prefix of the user, if it exists.

   userid

   is the ISPF shared pool variable ZUSER. This is typically the same as the user's TSO logonid.

   HFM2AUD

   is a constant that identifies the data set as an audit trail

   ssid

   is the Db2® subsystem ID

D*yymmdd*

is the date ZDT/Db2 created the audit trail data set

T*hhmmss*

is the time ZDT/Db2 created the audit trail data set

Note: If the AUDITHLQ option is set in HFM0POPT (see *Z Data Tools Customization Guide*), audit trail data sets have the name *audithlq*.HFMLOG.D*yymmdd*.T*hhmmss*.

If you created the audit trail data set yourself, use the data set name you specified in the supplied batch job HFMSMFX. For details, see the *Z Data Tools Customization Guide*.

2. To help identify your audit trail report, use the **Description** entry field. If the description contains embedded spaces, you must enclose it in quotation marks. The description you enter is displayed on the Formatted Audit Event Records section of the report.
3. Select the processing options you want:

   Print only changed columns

   Limits the amount of printed lines.

   Keep data set after printing

   Whether to keep or delete the data set after it is printed.

   Browse report

   The audit trail report is displayed using Print Browse (option 3.11).

   Note: The contents of the audit trail report are transferred to a SYSOUT class as specified by the **PRINTOUT** print option on the Set Print Processing Options panel. To browse the audit trail report, set the **PRINTOUT** print option to SYSOUT=*c*.

   Batch execution

   Generates JCL for printing the audit trail report using a batch job. For command syntax, see *"AUD (Print Audit Trail Report)"* in the Z Data Tools User's Guide and Reference.
4. Press Enter.

Audit Trail Report

The audit trail report is divided into two sections: Formatted Audit Events Records (Figure 81: Audit trail report: Formatted Audit Event Records section on page 337) and Summary Statistics (Figure 82: Audit trail report: Summary Statistics section on page 338).

Figure 81. Audit trail report: Formatted Audit Event Records section

```
------------------------------------------------------------------------------------------------------------------------
 HCL Z Data Tools Db2 Component           Audit Trail Report                                    Page 1
                                              Formatted Audit Event Records                     Date: 2004-09-2
                                                                                                Time: 11.14.39

Audit trail for EDIT function.
     Data set name: HFMUSER.HFM2AUD.DB26.D020918.T155156
 Audit description:
System Information:
     MVS System Id: FMD2
     Db2 Subsystem: DB26
     Current SQLID: HFMUSER
Db2 Object Information:
          Location:
             Owner: HFMUSER
              Name: EMP
Template Information:
     Data set name:
Audit event number: 1                  Data fetched on 02/09/18 at 15.52.00
     SQL statement:   SELECT "EMPNO", "FIRSTNME", "MIDINIT", "LASTNAME", "WORKDEPT", "PHONENO", "HIREDATE",
                       "JOB", "EDLEVEL", "SEX", "BIRTHDATE", "SALARY", "BONUS", "COMM" FROM "HFMUSER"."EMP"
------------------------------------------------------------------------------------------------------------------------
```

The first part of the Formatted Event Records section shows:

- The ZDT/Db2 function being audited (View, Edit, or Copy).
- The name of the data set holding the audit log records.
- The report description (if specified).
- Location information for the Db2® object being manipulated.

The second part of the Formatted Event Records section shows information for each audit event (each SQL call is defined as a separate audit event):

- The audit event number.
- A date and timestamp.
- The SQL statement.
- The Db2® column name.
- The type of key (if applicable).
- The original value of the column (if the column has been changed or deleted).
- The new value of the column (if the column has been changed or inserted).

If **Print only changed fields** was selected on the Print Audit Trail panel, only columns where the data has changed are shown.

Every ZDT/Db2 view, edit, or copy of a Db2® object is recorded as a separate session. If SMF recording is used, many sessions can be logged. Summary statistics are provided for each session, not the entire period being audited.

Figure 82. Audit trail report: Summary Statistics section

```
Summary Statistics
-------------------------------------------------------------------------------------------------------------------
 HCL Z Data Tools Db2 Component          Audit Trail Report                                    Page 2
                                             Summary Statistics                                Date: 2004-09-2
                                                                                               Time: 11.14.39

                       Data base opened on 2004-09-18 at 15.52.00 by userid: HFMUSER
                       Data base closed on 2004-09-18 at 15.55.47
   Total SQL statements: 1
          Rows inserted: 0
           Rows deleted: 0
          Rows replaced: 0
TOTAL Data base changes: 0
         Records bypassed: 0
Audit Trail Report Options
                Maximum Print Lines per Page: 60
      Print Only Changed Fields on an Update: NO
-------------------------------------------------------------------------------------------------------------------
```

The Summary Statistics section shows:

- The times the database was opened and closed.
- The user ID.
- Counts of the number of SQL calls, inserts, deletes, and row replacements for the ZDT/Db2 session.
- The total number of update events.

Related tasks

View and Edit options (options 1 and 2) on page 67
Setting options for the current ZDT/Db2 editor session on page 126
Copy utility options (option 3.3) on page 68
Viewing and changing Db2 data on page 122
Copying data from one Db2 object to another on page 246
Looking at the print output from your ZDT/Db2 session on page 333
Print settings (option 0.1) on page 64

Related references

Editor Options (1 of 8) panel on page 644
Print Audit Trail panel on page 761

Chapter 11. Managing Db2® privileges

Use the Privileges utility function to display, grant, and revoke Db2® privileges.

To perform these functions, you use the Manage Db2® Privileges panel.

The available functions for managing Db2® privileges are shown in Table 23: Privilege functions on page 339.

Table 23. Privilege functions

Privilege Type	Grant	Revoke
Database	Yes	Yes
Table space	Yes	Yes
Table/View	Yes	Yes
Column	Yes	No[23]
Plan	Yes	Yes
Package	Yes	Yes
Collection	Yes	Yes
Schema	Yes	Yes
Distinct type	Yes	Yes
Function	Yes	Yes
Stored procedure	Yes	Yes

To display the Manage Db2® Privileges panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `5` (Privileges)

Related tasks

Displaying privileges on page 342
Granting privileges on page 342
Revoking privileges on page 344

Related references

23. You cannot revoke privileges on specific columns. You must revoke the appropriate table privileges.

Manage Db2 Privileges panel on page 741
Primary Option Menu panel on page 760
Utility Functions panel on page 915

Using the Manage Db2® Privileges panel

The Manage Db2® Privileges panel consists of two sections:

Object Identification Criteria

Use this section of the panel to specify object identification criteria to identify the objects you want listed. ZDT/Db2 lists all objects, restricted by any such criteria, for which privileges exist. Input in this section depends on the **Object Type** you specify.

Location

Leave this field blank if the objects you want listed are at your current location. Otherwise, type the full name of the remote location, an asterisk (*) to display all available remote locations, or a pattern using wildcard characters. If you specify an asterisk on its own or a pattern, ZDT/Db2 displays a selection list showing the locations that match the input specification.

To select the location you want, type `s` against the required entry and press Enter.

Note: You can only specify a location if displaying privileges (command line blank).

Owner

To select objects regardless of their owner, leave this field blank. Otherwise, to restrict the list of objects by owner, type either the full name of the owner or a pattern using wildcard characters. If this entry field is not blank, ZDT/Db2 limits the objects listed to those matching the owner selection criteria you specify.

Note: You can only specify an owner for table or column object types.

Name

To select objects regardless of their name, leave this field blank. Otherwise, to restrict the list of objects by name, type either the full name of the object or a pattern using wildcard characters. If this entry field is not blank, ZDT/Db2 limits the objects listed to those matching the name selection criteria you specify.

In

This is an optional field you can use to qualify the following object types:

Object type	Qualify by specifying the…
Table space	Name of the database

Object type	Qualify by specifying the…
Column	Name of the table
Package	Name of the collection
Distinct type	Name of the schema
Function	Name of the schema
Stored procedure	Name of the schema

Object Type

Use this section to identify the type of Db2® object against which privilege operations are performed. The available functions for managing Db2® privileges are shown in Table 23: Privilege functions on page 339.

To select the object type you want, type the appropriate number in the **Object Type** selection field. The default value is 3 (table).

You select whether to display, grant, or revoke privileges by what, if anything, you type on the command line:

Display privileges

To select this processing option, leave the command line blank.

Grant privileges

To select this processing option, type `G` on the command line.

Revoke privileges

To select this processing option, type `R` on the command line.

Note:
To change an *existing* privilege, you must:

1. Revoke the existing privilege.
2. Grant the new privilege.

Related tasks

Using an asterisk (*) or a pattern in entry fields on page 50

Related references

Manage Db2 Privileges panel on page 741
Remote Db2 Location Selection panel on page 791

Displaying privileges

You can display privilege details for any of the object types listed on the Manage Db2® Privileges panel.

To display privilege details:

1. Type the required details (if any) in the **Object Identification Criteria** section.
2. Type the appropriate number in the **Object Type** selection field to select the object type you want, or use the default value, 3 (table).
3. Leave the command line blank.
4. Press Enter. The privilege display panel for the specified privilege types is displayed.

Related tasks

Granting privileges on page 342
Revoking privileges on page 344

Related references

Manage Db2 Privileges panel on page 741
Primary Option Menu panel on page 760
Utility Functions panel on page 915
Privileges panels on page 768

Granting privileges

You can grant (create new) privileges for any of the object types listed on the Manage Db2® Privileges panel.

To grant one or more privileges for an object, you can use any of the following methods:

- Method 1 (specifying the object using the grant privileges panel):
    1. On the Manage Db2® Privileges panel:
        a. Type the required details (if any) in the **Object Identification Criteria** section.
        b. Type the appropriate number in the **Object Type** selection field to select the object type you want, or use the default value, 3 (table).
    2. Type `G` on the command line.
    3. Press Enter.

       The grant privileges panel for the specified object type is displayed. The panel lists each privilege type, with a corresponding selection entry field, that can apply to the object type.

       On the grant privileges panel:

        a. Select the privileges you want to grant by entering *"Y"* or *"G"* in the corresponding selection fields.
        b. In the entry field (or fields) provided, specify the name of the object.

c. In the **To** entry field, specify the SQLID of the user to whom the privileges are to be granted (the grantee).
d. Press Enter.

- Method 2 (specifying the object from a privilege display panel):

Note: This method only works if there are already privileges for the Db2® object. If not, ZDT/Db2 issues the message “No rows selected”.

1. On the Manage Db2® Privileges panel,
   a. Type the required details (if any) in the **Object Identification Criteria** section.
   b. Type the appropriate number in the **Object Type** selection field to select the object type you want, or use the default value, 3 (table).
2. Leave the command line blank.
3. Press Enter. The privilege display panel for the specified object type is displayed listing the selected objects.
4. On the privilege display panel, type `G` in the **Cmd** field against the object for which you want to grant privileges.
5. Press Enter. The grant privileges panel for the specified object is displayed.

Note:
If you use this method to navigate to the grant privileges panel for the specified object type, the following details are pre-entered:
- The existing setting for each privilege type (Y, G, or blank)
- The name details for the object
- The SQLID of the user to whom the existing privileges are granted (the grantee)

On the grant privileges panel:

a. Select the privileges you want to grant by entering *“Y”* or *“G”* in the corresponding selection fields.
b. Press Enter.

- Method 3 (using the Object List utility):
1. Use the utility function, Object List (option 3.4), to display a list of the type of objects you want.
2. On the object list panel for the selected object type, type `G` in the **Cmd** field against the object for which you want to grant privileges.
3. Press Enter.

   The grant privileges panel for the specified object type is displayed (as for Method 1). The panel lists each privilege type, with a corresponding selection entry field, that can apply to the object type.
4. Follow the remaining steps as for Method 1 (3.a on page 342 to 3.d on page 343)

If the grant privilege processing is successful, ZDT/Db2 displays a message indicating that the grant ran successfully. Otherwise, a panel is displayed showing the formatted SQL return code.

Related tasks

Displaying privileges on page 342
Revoking privileges on page 344
Displaying or printing a list of objects on page 304

Related references

Manage Db2 Privileges panel on page 741
Primary Option Menu panel on page 760
Utility Functions panel on page 915
Grant privileges panels on page 715

Revoking privileges

You can revoke (remove existing) privileges for any of the object types listed on the Manage Db2® Privileges panel apart from column. (You cannot revoke privileges on specific columns. You must revoke the appropriate table privileges.)

To revoke one or more privileges for an object, you can use either of the following methods:

- Method 1 (specifying the object using the revoke privileges panel):
    1. On the Manage Db2® Privileges panel,
        a. Type the required details (if any) in the **Object Identification Criteria** section.
        b. Type the appropriate number in the **Object Type** selection field to select the object type you want, or use the default value, 3 (table).
    2. Type `R` on the command line.
    3. Press Enter.

       The revoke privileges panel for the specified object type is displayed. If there is more than one type of privilege that can apply to the object type, the panel lists each privilege type with a corresponding entry field.

       On the revoke privileges panel:

        a. Select the privileges you want to revoke by entering *any character* in the corresponding selection fields.

           Note: For object types where there is only one type of privilege, this step does not apply.

        b. In the entry field provided (for some types of privilege there are two entry fields), specify the name of the object.
        c. In the **From** entry field, specify the SQLID of the user for whom the privileges are to be revoked (the revokee).
        d. In the **By** entry field, specify the SQLID of the user who granted the privileges (the grantor), or *"ALL"* for all grantors.
        e. Press Enter.
- Method 2 (specifying the object from a privilege display panel):

1. On the Manage Db2® Privileges panel,
   a. Type the required details (if any) in the **Object Identification Criteria** section.
   b. Type the appropriate number in the **Object Type** selection field to select the object type you want, or use the default value, 3 (table).
2. Leave the command line blank.
3. Press Enter. The privilege display panel for the specified object type is displayed listing the selected objects.
4. On the privilege display panel, type `R` in the **Cmd** field against the object for which you want to revoke the privileges.
5. Press Enter. The revoke privileges panel for the specified object is displayed.

   Note:
   If you use this method to navigate to the revoke privileges panel (for the specified object type), the following details are pre-entered:
   - The existing setting for each privilege type (Y, G, or blank)
   - The name details for the object
   - The SQLID of the user for whom the privileges are to be revoked (the revokee) in the **From** entry field.

   On the revoke privileges panel:

   a. Select the privileges you want to revoke by entering *any character* in the corresponding selection fields.

      Note: For object types where there is only one type of privilege, this step does not apply.

   b. In the **By** entry field, specify the SQLID of the user who granted the privileges (the grantor), or *"ALL"* for all grantors.
   c. Press Enter.

If the revoke privilege processing is successful, ZDT/Db2 displays a message indicating that the revoke ran successfully. Otherwise, a panel is displayed showing the formatted SQL return code.

Related tasks

Displaying privileges on page 342
Granting privileges on page 342

Related references

Manage Db2 Privileges panel on page 741
Primary Option Menu panel on page 760
Utility Functions panel on page 915
Revoke privileges panels on page 798

Chapter 12. Generating batch JCL for Db2® utility jobs

To generate batch JCL to execute the Db2® utility jobs listed below, use the Utilities utility function (3.9). You do not need to know the utility control statements for these operations.

- COPY
- LOAD
- REBUILD
- RECOVER
- REORG
- RUNSTATS
- UNLOAD

For an explanation of the uses of these Db2® utility jobs, refer to the *Db2® Utility Guide and Reference*.

Note: To successfully execute the generated utility job, you must have the appropriate Db2® privileges on any Db2® objects that are referenced.

The main panel for the Utilities utility is the Db2® Utilities panel.

To display the Db2® Utilities panel,

1. From the Primary Option Menu panel, enter `3` (Utilities) to display the Utility Functions panel.
2. From the Utility Functions panel, enter `9` (Utilities)

Use the fields on the Db2® Utilities panel to select:

- The type of Db2® utility you want to run
- The type of Db2® object that the utility processes
- Details of the particular Db2® objects you want to run a utility on

When you press Enter, all objects at the local Db2® server are displayed, unless you entered information in the *"Db2® Object Details"* fields on the Db2® Utilities panel in which case the displayed rows are limited to the objects that match the information you have specified.

The JCL for the Db2® utility job is generated and presented in an ISPF edit session. You can make any required changes prior to submitting the job, or use the ISPF editor commands to save a copy of the JCL for later use.

Table 24: Db2 utilities you can use with Db2 objects on page 347 show which Db2® utilities you can use with which Db2® objects.

Table 24. Db2® utilities you can use with Db2® objects

Db2® utility	Use with these Db2® objects				
	1. Tables	2. Table spaces	3. Index spaces	4. Indexes	5. Indexes for table space
1. COPY		Yes (see COPY (table spaces) on page 355)	Yes (see COPY (index spaces) on page 356)		
2. LOAD	Yes (see LOAD utility on page 357)				
3. REBUILD				Yes (see REBUILD utility (indexes) on page 366)	Yes (see REBUILD utility (indexes for table spaces) on page 367)
4. RECOVER		Yes (see RECOVER (table spaces) on page 368)	Yes (see RECOVER (index spaces) on page 370)		
5. REORG		Yes (see REORG (table spaces) on page 372)		Yes (see REORG (indexes) on page 374)	
6. RUNSTATS		Yes (see RUNSTATS (table spaces) on page 377)		Yes (see RUNSTATS (indexes) on page 378)	
7. UNLOAD	Yes (see UNLOAD (tables) on page 381)	Yes (see UNLOAD (table spaces) on page 386)			

Related tasks

COPY utility on page 354
LOAD utility on page 357
REBUILD utility on page 365
RECOVER utility on page 368
REORG utility on page 371
RUNSTATS utility on page 376
UNLOAD utility on page 380
Generating a Db2 utility job on page 350

Related references

Db2 Utilities panel on page 627
Primary Option Menu panel on page 760
Utility Functions panel on page 915

Setting options for Db2® utilities

You can set Db2® utility options to:

- Insert JCL DD statements for specified LISTDEF or TEMPLATE libraries
- Modify the Db2® utility functions to include DD name clauses with specified template names
- Insert LISTDEF, TEMPLATE, and OPTIONS Db2® utility statements

To change the option settings for Db2® utilities, first select **0 (Settings)** from the Primary Option menu to display the Set Processing Options panel. The new settings apply for the current subsystem only. When you reconnect to the same subsystem, your option settings are retrieved.

To reset the option settings back to the installation defaults (for the current Db2® subsystem), use the RESET primary command or the Reset function key (F6).

The Set Processing Options panel shows menu items for:

Unload utility options

The ZDT/Db2 Unload Utility options apply when the UNLOAD utility is used to transfer data.

Options utility options

The ZDT/Db2 Options utility options assist in generating Db2® OPTIONS utility control statements.

Listdef utility options

The ZDT/Db2 Listdef utility options assist in generating Db2® LISTDEF utility control statements and in generating JCL for utility batch jobs to access a LISTDEF library.

Template utility options

The ZDT/Db2 Template utility options assist in generating Db2® TEMPLATE utility control statements and in generating JCL for utility batch jobs to access a template library. Also, options are provided that allow you to specify template names for DD names clauses of various Db2® utility control statements.

For a full description of Db2® utilities, see the *Db2® Utility Guide and Reference*.

Before you generate batch JCL to execute a Db2® Utility, check that these options are set to the values you want.

ZDT/Db2 uses the above options to generate JCL that refers to the appropriate LISTDEF and TEMPLATE libraries and also, where applicable, to generate an OPTIONS, LISTDEF, or TEMPLATE utility statement in the utility batch job. If specified, ZDT/Db2 places the OPTIONS, LISTDEF, and TEMPLATE statements in the SYSIN deck before the utility statement that corresponds to the utility selected from the Db2® Utilities panel.

To change the option settings for a Db2® utility:

1. From either the Set Processing Options panel or the **Options** pull-down menu, select the menu item for the options you want to change:

   11

   Unload

   12

   Options

   13

   Listdef

   14

   Template

   ZDT/Db2 displays the selected options panel.
2. Change the option settings as necessary

Related tasks

Setting default processing options on page 62

Related references

Set Processing Options panel on page 832
Db2 Utilities panel on page 627
UNLOAD Options panel on page 905
OPTIONS Options panel on page 750
LISTDEF Options panel on page 729
Additional LISTDEF Statements panel on page 449

TEMPLATE Options (1 of 2) panel on page 897
TEMPLATE Options (2 of 2) panel on page 898
Additional TEMPLATE Statements panel on page 450
RESET primary command on page 994

Generating a LISTDEF statement in the Db2® utility job

The Db2® Utilities panel shows a **Specification** section and a **LISTDEF name** entry field. You can use these entry fields to specify the name of a LISTDEF.

If you select LISTDEF in the **Specification** section, you must also specify a LISTDEF name as the object of the utility.

To use a LISTDEF name as the object of the utility:

1. In the **Specification** section, select **2. LISTDEF**
2. Enter the name of the LISTDEF in **LISTDEF name**

Note: The LOAD syntax does not have a LISTDEF clause. If you select LOAD in the **Db2® Utility** section and LISTDEF in the **Specification** section, ZDT/Db2 issues an error message.

Related references

Db2 Utilities panel on page 627

Generating a Db2® utility job

To generate a Db2® utility job:

1. Specify the Db2® utility you want to generate:

   1

   COPY

   2

   LOAD

   3

   REBUILD

   4

   RECOVER

   5

   REORG

6

RUNSTATS

7

UNLOAD

2. Specify the object type:

1

Tables

2

Table spaces

3

Index spaces

4

Indexes

5

Indexes for table spaces

3. For COPY, REBUILD, RECOVER, REORG, RUNSTATS, and UNLOAD (that is, all the ZDT/Db2 Db2® utilities except LOAD), you can specify that a LIST clause is to be included in the JCL generated by ZDT/Db2 for the nominated Db2® utility.

   To generate a LIST clause in the JCL for the Db2® utility, select **2. LISTDEF**; otherwise, select **1. Object name**.

4. You now need to supply details about either the object or the LISTDEF for the Db2® utility job:
   - If you specified **1. Object name** in the Specification section (Step 3), optionally, specify details about the object by using one or more of the four filter fields:
     - **Database**
     - **Table/Index space**
     - **Owner** (Table or Index)
     - **Name** (Table or Index)

     ZDT/Db2 validates any details you enter and carries them forward to the next panel.

     For each of the preceding filter fields, you can enter:
     - The full name of the item.
     - An asterisk (*) to display a list of items from which you can select the one you want.

       If you enter an asterisk (*) in any of the four filter fields, ZDT/Db2 displays the respective selection list panel from which you can select a value to put into the filter field. The records displayed in these panels are restricted by any other related filter you have previously entered. Each combination of utility and Db2® object results in certain valid filter combinations. Incorrect filter combinations are rejected.

- A pattern to display a list of all items whose name matches the pattern specification.
- Blanks. No filtering effect occurs.

◦ If you specified **2. LISTDEF** in the Specification section (Step 3), enter the name to be used in the LIST clause in **LISTDEF name**.

5. Press Enter.

   ZDT/Db2 displays the panel for the specified Db2® utility and object type.

   If you specified **1. Object name** in the Specification section (Step 3), the panel shows the qualifying list of rows, with various input and output fields. Select the rows you want and change the values in any of the editable fields as required.

   If you specified **2. LISTDEF** in the Specification section (Step 3), ZDT/Db2 displays an alternative panel for the Db2® utility. See Working with LISTDEF utility panels on page 353.

6. Press Enter.

   ZDT/Db2 displays the generated JCL which you can edit if you need to before you submit it as a job. (The LOAD and UNLOAD utilities are exceptions to this rule, as both utilities show another series of panels, before displaying the generated JCL.) Figure 83: Sample JCL generated for COPY Utility (table spaces) on page 352 shows an example of generated JCL.

   Figure 83. Sample JCL generated for COPY Utility (table spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** **************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP COPY1: COPY TABLESPACE DSN8D81A.DSN8S81D
000010 //*********************************************************************
000011 //COPY1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSCOPY  DD DISP=(NEW,CATLG,DELETE),
000015 //          DSN=HFMUSER.DSN8D81A.DSN8S81D.F00000.D2601115,
000016 //          UNIT=SYSALLDA,SPACE=(TRK,(1,1),RLSE)
000017 //DSNUPROC.SYSIN    DD *
000018 COPY TABLESPACE DSN8D81A.DSN8S81D
000019   COPYDDN(SYSCOPY)
000020 /*
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

7. To submit the JCL as a job stream for batch processing,

   use the SUBMIT command.

The following sections describe each of the Db2® utilities that ZDT/Db2 supports.

Related tasks

Using an asterisk (*) or a pattern in entry fields on page 50
COPY utility on page 354
LOAD utility on page 357
REBUILD utility on page 365
RECOVER utility on page 368
REORG utility on page 371
RUNSTATS utility on page 376
UNLOAD utility on page 380

Working with LISTDEF utility panels

When you specify **2. LISTDEF** in the Specification section and the name of the LISTDEF in **LISTDEF name** of one of the Db2® utility panels (excluding LOAD), ZDT/Db2 displays an alternative panel showing the name of the LISTDEF. You use the LISTDEF form of the utility panel to specify one or more names to be used in the LIST clause in the JCL generated for the Db2® utility.

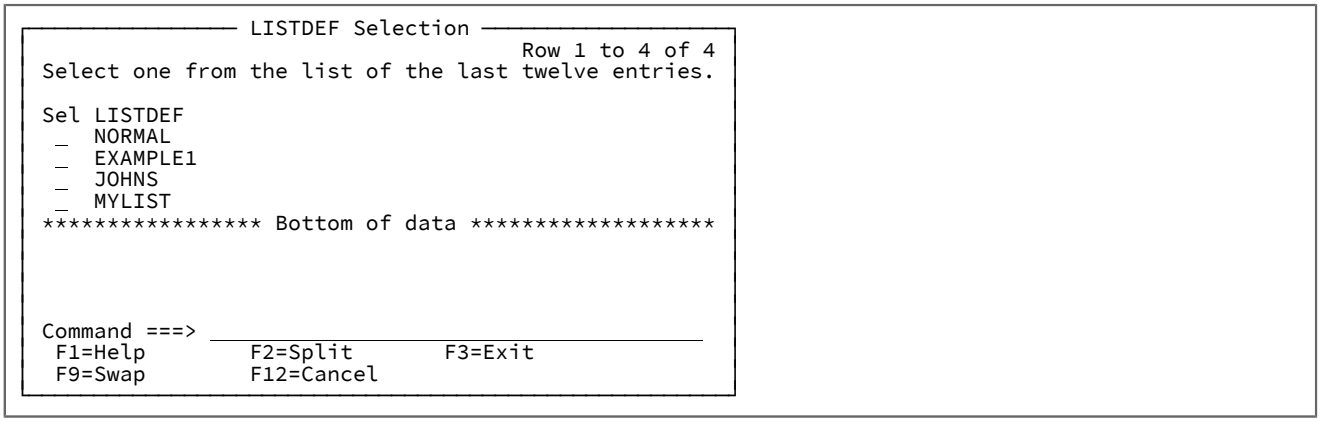
Initially the panel shows a single line containing the name you specified in the **LISTDEF name** field on the Db2® utility panel.

You can manipulate the panel rows as necessary by entering line commands in the **CMD** field to insert, repeat, delete, or select rows.

An asterisk (*) in the **PRC** column indicates the LISTDEF in the row is selected. Use the S command to toggle unselected rows to selected and selected rows to unselected.

You cannot select a row where the **LISTDEF Name** field is blank.

To display the LISTDEF Selection pop-up panel, type an asterisk (*) in the **LISTDEF Name** field.



Select a LISTDEF name from the displayed list by entering an s in the **Sel** field.

To create a utility batch job with the selected LISTDEFs, press Enter.

For more information about the LISTDEF name, see the LIST keyword in the *Db2 Utility Guide and Reference*.

COPY utility

To select the COPY utility, enter `1` in the **Db2® utility** input field on the Db2® Utilities panel.

If you selected *"Table spaces"* in the **Object Type** section of the Db2® Utilities panel, the COPY Utility (Table spaces) panel is displayed.

If you selected *"Index spaces"* in the **Object Type** section of the Db2® Utilities panel (and did not specify name of a LISTDEF), one of two forms of the COPY Utility (Index spaces) panel is displayed. The format of the panel depends on which, if any, of the filter fields you specified on the Db2® Utilities panel.

Note: If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the COPY Utility (Table spaces) panel or COPY Utility (Index spaces) panel.

Related tasks

Default values on page 354
COPY (table spaces) on page 355
COPY (index spaces) on page 356

Related references

Db2 Utilities panel on page 627
COPY Utility (Table Spaces) panel on page 524
COPY Utility (Table Spaces) with LISTDEF panel on page 526
COPY Utility (Index Spaces) panel on page 517
COPY Utility (Index Spaces) with LISTDEF panel on page 519
TEMPLATE Options (2 of 2) panel on page 898

Default values

The default naming conventions used for image copy data sets in the generated JCL are:

- &ZUSER..&DBNAME..&OBJNAME..&PRE&PARTNO..D&HFM2DTIM.
- &ZUSER, &DBNAME, and &PARTNO are self-explanatory.
- &OBJNAME is either the table space name or the index space name.
- &PRE can be:

  F

  Full

  I

  Incremental

C

Change Limit. This can result in Db2® performing a full or incremental imagecopy.

- &HFM2DTIM contains *jjjhhmm*, where with *jjj* is the Julian day, and *hhmm* is the time.

  For example, OSPEED.UTILDB1.T1RIX1.F00000.D1011044

COPY (table spaces)

This option corresponds to the COPY TABLE SPACE form of the COPY utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name** or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select a table space from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.
  - You can modify the COPY utility statement by changing the values in the following input fields:
    - FCP
    - Change Limits
    - RPO
  - Figure 83: Sample JCL generated for COPY Utility (table spaces) on page 352 shows an example of generated JCL for copying a table space.
- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:

- ZDT/Db2 displays an alternative form of the COPY Utility (Table Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the COPY statement.
- Figure 84: Sample JCL generated for COPY Utility (table spaces) with LISTDEF specified on page 356 shows an example of generated JCL for copying a table space where a LISTDEF has been specified.

Figure 84. Sample JCL generated for COPY Utility (table spaces) with LISTDEF specified

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP COPY1: COPY TABLESPACE LIST LDEFNAME
000010 //*********************************************************************
000011 //COPY1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSCOPY  DD DISP=(NEW,CATLG,DELETE),
000015 //          DSN=HFMUSER.LDEFNAME.F.D2601124,
000016 //          UNIT=SYSALLDA,SPACE=(TRK,(1,1),RLSE)
000017 //DSNUPROC.SYSIN    DD *
000018 COPY LIST LDEFNAME
000019   COPYDDN(SYSCOPY)
000020 /*
****** ************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related references

COPY Utility (Table Spaces) panel on page 524
COPY Utility (Table Spaces) with LISTDEF panel on page 526
Db2 Utilities panel on page 627

COPY (index spaces)

This option corresponds to the COPY INDEXSPACE form of the COPY utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - When you use COPY (index spaces), you can use the following combinations of filter fields on the Db2® Utilities panel:
    - **Database name**, or **Table/index space name**.
    - **Database name**, **Table/index owner**, or **Table/index name**.
  - If you entered information in any of the preceding filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select an index space from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.
  - The COPY INDEXSPACE statements generated by ZDT/Db2 specify the database and index space name (rather than the index owner and index name), regardless of the way you specified the index to be copied.

Figure 85: Sample JCL generated for COPY Utility (index spaces) on page 357 shows an example of generated JCL for copying an index space.

Figure 85. Sample JCL generated for COPY Utility (index spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP COPY1: COPY INDEXSPACE JLV3D81A.XEMP2
000010 //*********************************************************************
000011 //COPY1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSCOPY  DD DISP=(NEW,CATLG,DELETE),
000015 //          DSN=HFMUSER.JLV3D81A.XEMP2.F00000.D2601130,
000016 //          UNIT=SYSALLDA,SPACE=(TRK,(1,1),RLSE)
000017 //DSNUPROC.SYSIN    DD *
000018 COPY INDEXSPACE JLV3D81A.XEMP2
000019   COPYDDN(SYSCOPY)
000020 /*
****** **************************** Bottom of Data ****************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:
  - ZDT/Db2 displays an alternative form of the COPY Utility (Index Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the COPY statement.

Related references

COPY Utility (Index Spaces) panel on page 517
COPY Utility (Index Spaces) with LISTDEF panel on page 519
Db2 Utilities panel on page 627

LOAD utility

This option corresponds to the Db2® LOAD utility control statement.

To select the LOAD utility, on the Db2® Utilities panel:

1. Enter 2 (LOAD) in the **Db2® Utility** entry field.
2. Enter 1 (Tables) in the **Object Type** entry field.

The LOAD Utility panel is displayed.

Note:
You can also use the Import (option 3.6) utility to load data from a QSAM or VSAM data set into a Db2® table, either in foreground or in batch.



Related tasks

Copying data from a VSAM or QSAM file on page 251
Input data on page 358
Loading data using LOAD utility control statements on page 359
Loading data described by a copybook or template on page 361

Related references

Db2 Utilities panel on page 627
LOAD Utility panel on page 733
TEMPLATE Options (2 of 2) panel on page 898

Input data

The ZDT/Db2 process that builds the LOAD utility JCL requires the input data to be in a fixed-length format, as it builds fixed-position field specification statements.

You can use the LOAD utility where the data to be loaded is:

- Described by existing LOAD utility control statements.

  For example, the REORG utility can be used to unload Db2® data into Db2® *"unload"* format by specifying the UNLOAD EXTERNAL option. The REORG utility also produces a file containing Db2® LOAD utility control statements to (re)load the data into the original tables.
- Described by a copybook or template.

  You must specify the name of a copybook or ZDT/Db2 template that describes the input data. The format of the data needs to be consistent with the requirements of the LOAD utility.

Related tasks

REORG utility on page 371
Setting options for Db2 utilities on page 348
Loading data using LOAD utility control statements on page 359
Loading data described by a copybook or template on page 361

Loading data using LOAD utility control statements

Note: To load data in this way, the default INDDN name (in the **INDDN template name** field on the second TEMPLATE Options panel) must be blank.

To load data from a previous Db2® *"unload"* operation:

1. In the entry fields for input data, specify the name of the data set (**Data set name**), and optionally the name of the member (**Member**), containing the data to be loaded.
2. In the entry fields for utility control statements, specify the name of the data set (**Data set name**), and optionally the name of the member (**Member**), containing the utility control statements.
3. Select option 1 (*"The data is described by utility control statements"*)
4. Press Enter.

ZDT/Db2 constructs a batch job to run the Db2® LOAD utility job using the input details you have specified. Figure 86: Sample JCL generated for LOAD Utility using LOAD utility control statements on page 360 shows an example of generated JCL for loading a table using data in Db2® "unload" format.

Figure 86. Sample JCL generated for LOAD Utility using LOAD utility control statements

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** **************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //*
000010 //* ZDT/Db2 GENERATED JOB TO LOAD A TABLE
000011 //*
000012 //*********************************************************************
000013 //*
000014 //*********************************************************************
000015 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000016 //*********************************************************************
000017 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000018 //*
000019 //*********************************************************************
000020 //* STEP LOAD: LOAD THE TABLE
000021 //*********************************************************************
000022 //LOAD     EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000023 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000024 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000025 //DSNUPROC.SYSREC00 DD DISP=SHR,              INPUT DATA
000026 //          DSN=HFMUSER.DATA(HFMCDATA)
000027 //DSNUPROC.SYSDISC  DD SYSOUT=*               DISCARDED RECORDS
000028 //DSNUPROC.SYSERR   DD UNIT=SYSALLDA,
000029 //          SPACE=(TRK,(1,1))                 ERROR INFORMATION
000030 //DSNUPROC.SYSMAP   DD UNIT=SYSALLDA,
000031 //          SPACE=(TRK,(1,1))                 INTERNAL MAPPING DATA SET
000032 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000033 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000034 //DSNUPROC.SORTOUT  DD UNIT=SYSALLDA,
000035 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000036 //DSNUPROC.SYSIN    DD *
000037 LOAD DATA
000038 PREFORMAT
000039 LOG YES
000040 INDDN SYSREC00
000041 RESUME YES
000042 FLOAT(IEEE)
000043 ASCII
000044 SORTKEYS 100
000045 SORTDEVT SYSDA
000046 INTO TABLE
000047 "HFMUSER"."EMP"
000048 (
000049  "EMPNO             " POSITION(1     :2     ) CHAR(6)
000050 ,"SALARY            " POSITION(27    :30    ) DECIMAL
000051 )
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Loading data described by a copybook or template on page 361

Related references

LOAD Utility panel on page 733

Loading data described by a copybook or template

You can use the LOAD utility to load data in a sequential data set, providing a copybook or template exists that describes the format of the input data.

In this case, you use a series of panels:

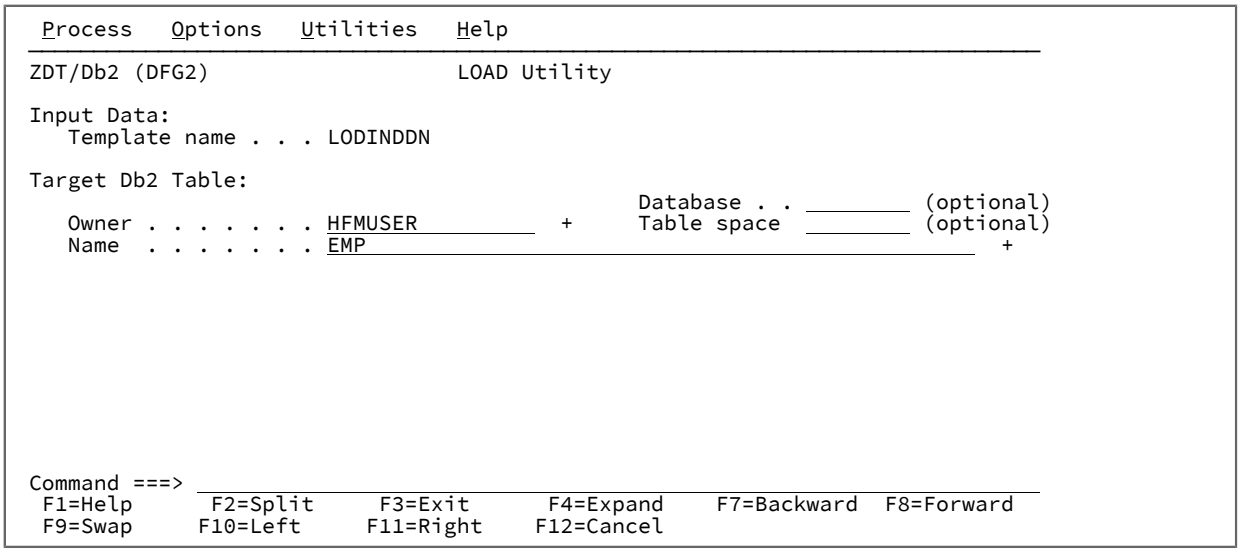
- On the LOAD Utility panel, you specify the name of the data set containing the input data, and the name of the target table.
- On the second panel, you specify the *"From"* copybook or template that describes the input data.
- On the third panel, you can specify the *"To"* template that describes the target table. Alternatively, you can use ZDT/Db2 to generate a template for the target table using the information in the Db2® catalog.

  At this stage, you can define the mapping between fields in the input template and fields in the output template.

To load data described by a copybook or template:

1. In the entry fields for input data, specify the name of the data set (**Data set name**), and optionally the name of the member (**Member**), containing the data to be loaded.
2. In the entry fields for the target Db2® table (**Table owner**, **Table name**, and optionally **Database** and **Table space**), specify the name of the table into which the data is to be loaded. In any of these fields, you can enter either an asterisk (*) to display a selection list, or wildcards.
3. Select option 2 (*"The data is described by a template or copybook"*)
4. Press Enter.

   If you have specified a default name for an INDDN clause (in the **INDDN template name** field on the second TEMPLATE Options panel), ZDT/Db2 displays an interim panel showing the default name (for example, LODINDDN) in the **Template name** field for the input data:



ZDT/Db2 displays a panel where you can specify the input template or copybook.

5. Specify the *"From"* template details. Perform one of these actions:
   - Specify a copybook or template:
     - In the entry fields for the input template or copybook, specify the name of the data set (**Data set name**), and optionally the name of the member (**Member**), for the template or copybook that describes the format of the input data.
     - Select 1 (Above) for the **Template usage** option.
   - Use the last (previously used) template:
     - Select 2 (Previous) for the **Template usage** option.
6. To edit the template for the input data before proceeding to the LOAD utility *"To"* template panel, select the **View template** option. You can select this option by entering either a / or an "A".

   Note: The template edit operations have no effect when the template is used to generate Db2® LOAD utility control statements. If you select this option, you cannot update and save changes to the template.

7. Press Enter.

   ZDT/Db2 displays a panel where you can specify the template for the target Db2® table.

8. Specify the *"To"* template details using a combination of the **To Template** and **Template usage** entry fields. You can:
   - Specify a template that describes the format of the input data:
     - Use **Data set name**, and optionally **Member**, to specify the template.
     - Select 1 (Above).
   - Use the last (previously used) template:
     - Select 2 (Previous).
   - Use ZDT/Db2 to generate a template using the information in the Db2® catalog:
     - Select 3 (Generate from table).

- Use ZDT/Db2 to generate a template using the information in the Db2® catalog and store the generated template using the name you specify (if the member already exists, ZDT/Db2 replaces it):
  - Use **Data set name**, and optionally **Member**, to specify the name that you want ZDT/Db2 to use when storing the generated template.
  - Select 4 (Generate/Replace).

9. To change the options for the current LOAD session,

   select **Edit load utility options** by entering either a / or an “A”.

   ZDT/Db2 displays the Load Utility Options panel for editing.

   Note: Any options you set on this panel apply for the current LOAD session only. When you exit from the LOAD session, the load options revert to the global LOAD options.

10. To map the columns (or fields) in the “From” template to the columns in the “To” template, select **Edit load utility options** by entering either a / or an “A”. ZDT/Db2 displays the Template Mapping panel.
11. Press Enter.

ZDT/Db2 constructs a batch job to run the Db2® LOAD utility job using the input details you have specified. Figure 87: Sample JCL generated for LOAD Utility using data described by a copybook or template (continued in next figure) on page 364 shows an example of generated JCL for loading a table using data described by a copybook or template.

Figure 87. Sample JCL generated for LOAD Utility using data described by a copybook or template (continued in next figure)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //*
000010 //* ZDT/Db2 GENERATED JOB TO LOAD A TABLE
000011 //*
000012 //*********************************************************************
000013 //*
000014 //*********************************************************************
000015 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000016 //*********************************************************************
000017 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000018 //*
000019 //*********************************************************************
000020 //* STEP LOAD: LOAD THE TABLE
000021 //*********************************************************************
000022 //LOAD     EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000023 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000024 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000025 //DSNUPROC.SYSREC00 DD DISP=SHR,              INPUT DATA
000026 //          DSN=HFMUSER.DATA(HFMCDATA)
000027 //DSNUPROC.SYSDISC  DD SYSOUT=*               DISCARDED RECORDS
000028 //DSNUPROC.SYSERR   DD UNIT=SYSALLDA,
000029 //          SPACE=(TRK,(1,1))                 ERROR INFORMATION
000030 //DSNUPROC.SYSMAP   DD UNIT=SYSALLDA,
000031 //          SPACE=(TRK,(1,1))                 INTERNAL MAPPING DATA SET
000032 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000033 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000034 //DSNUPROC.SORTOUT  DD UNIT=SYSALLDA,
000035 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000036 //DSNUPROC.SYSIN    DD *
000037 LOAD DATA
000038 PREFORMAT
000039 LOG YES
000040 INDDN SYSREC00
000041 RESUME YES
000042 FLOAT(IEEE)
000043 ASCII
000044 SORTKEYS 100
000045 SORTDEVT SYSDA
000046 INTO TABLE
000047 "HFMUSER"."EMP"
000048 (
000049  "EMPNO             " POSITION(1     :2     ) CHAR(6)
000050 ,"SALARY            " POSITION(27    :30    ) DECIMAL
000051 )
****** ***************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Loading data using LOAD utility control statements on page 359
Mapping data on page 112
Selecting options on ZDT/Db2 panels on page 45
Editing a template on page 84
Db2 LOAD utility options (option 3.L) on page 69
Using an asterisk (*) or a pattern in entry fields on page 50

Related references

LOAD Utility panel on page 733
LOAD Utility - Using Templates panel on page 735
LOAD from panel on page 731
LOAD Utility Options panel on page 737
Template Mapping panel on page 900
TEMPLATE Options (2 of 2) panel on page 898

REBUILD utility

This option corresponds to the Db2® REBUILD INDEX utility control statement.

To select the REBUILD utility, enter `3` in the **Db2® Utility** input field on the Db2® Utilities panel.

If you selected Table spaces in the **Object Type** section of the panel, the REBUILD Utility (Indexes) panel is displayed.

If you selected Indexes or Index spaces in the **Object Type** section of the panel, the REBUILD Utility (Indexes for Table Spaces) panel is displayed.

Note: If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the REBUILD Utility (Indexes) panel or REBUILD Utility (Indexes for Table Spaces) panel.

Related tasks

REBUILD utility (indexes) on page 366
REBUILD utility (indexes for table spaces) on page 367

Related references

Db2 Utilities panel on page 627
REBUILD Utility (Indexes) panel on page 777
REBUILD (Indexes) with LISTDEF panel on page 779
REBUILD (Indexes for Table Spaces) panel on page 773
REBUILD (Indexes for Table Spaces) with LISTDEF panel on page 775

REBUILD utility (indexes)

To include a WORKDDN clause in the generated REBUILD statement, specify the required template name in **WORKDDN template name** on the second TEMPLATE Options panel.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name**, **Table/index owner**, or **Table/index name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select an index from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected indexes.
  - Figure 88: Sample JCL generated for REBUILD Utility (indexes) on page 366 shows an example of generated JCL for rebuilding indexes.

    In this example, ZDT/Db2 has generated two JCL steps, one for each index selected from the displayed rows on the previous panel.

    Figure 88. Sample JCL generated for REBUILD Utility (indexes)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                        Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000010 //*********************************************************************
000011 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000012 //*
000013 //*********************************************************************
000014 //* STEP RBIX1: REBUILD INDEX
000015 //* "HFMUSER"."XACT2"
000016 //*********************************************************************
000017 //RBIX1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000018 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000019 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000020 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000021 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000022 //DSNUPROC.SYSIN    DD *
000023 REBUILD INDEX (
000024 "HFMUSER"."XACT2"
000025 )
000026    WORKDDN(SYSUT1) SORTDEVT SYSALLDA SORTNUM 4 SORTKEYS
000027 /*
****** ***************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:
  - ZDT/Db2 displays an alternative form of the REBUILD Utility (Indexes) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the REBUILD statement.

Related tasks

REBUILD utility (indexes for table spaces) on page 367

Related references

Db2 Utilities panel on page 627
REBUILD Utility (Indexes) panel on page 777
REBUILD (Indexes) with LISTDEF panel on page 779
TEMPLATE Options (2 of 2) panel on page 898

REBUILD utility (indexes for table spaces)

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name** or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select a table space from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.
  - Figure 89: Sample JCL generated for REBUILD Utility (indexes for table spaces) on page 367 shows an example of generated JCL for rebuilding indexes for table spaces.

  Figure 89. Sample JCL generated for REBUILD Utility (indexes for table spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*******************************************************************
000009 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000010 //*******************************************************************
000011 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000012 //*
000013 //*******************************************************************
000014 //* STEP RBIX1: REBUILD INDEX ALL FOR DSN8D81A.DSN8S81D
000015 //*******************************************************************
000016 //RBIX1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000017 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000018 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000019 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000020 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000021 //DSNUPROC.SYSIN    DD *
000022 REBUILD INDEX (ALL) TABLESPACE DSN8D81A.DSN8S81D
000023    WORKDDN(SYSUT1) SORTDEVT SYSALLDA SORTNUM 4 SORTKEYS
000024 /*
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:
  - ZDT/Db2 displays an alternative form of the REBUILD (Indexes for Table Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the REBUILD statement.

Related tasks

REBUILD utility (indexes) on page 366

Related references

Db2 Utilities panel on page 627
REBUILD (Indexes for Table Spaces) panel on page 773
REBUILD (Indexes for Table Spaces) with LISTDEF panel on page 775

RECOVER utility

The RECOVER utility option generates JCL to run the Db2® RECOVER utility job.

To select the RECOVER utility, enter `4` in the **Db2® Utility** input field on the Db2® Utilities panel.

If you selected Table spaces in the **Object Type** section of the panel, the RECOVER Utility (Table spaces) panel is displayed.

If you selected Index spaces in the **Object Type** section of the panel, the RECOVER Utility (Index spaces) panel is displayed.

Note: If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the RECOVER Utility (Table Spaces) panel or RECOVER Utility (Index Spaces) panel.

Related tasks

RECOVER (table spaces) on page 368
RECOVER (index spaces) on page 370

Related references

Db2 Utilities panel on page 627
RECOVER Utility (Table Spaces) panel on page 783
RECOVER Utility (Table Spaces) with LISTDEF panel on page 785
RECOVER Utility (Index Spaces) panel on page 780
RECOVER Utility (Index Spaces) with LISTDEF panel on page 782

ZDT/Db2 only supports the following form of the RECOVER utility statement:

```
RECOVER object TOCOPY datasetname TOVOLUME CATALOG
```

RECOVER (table spaces)

This option corresponds to the RECOVER TABLE SPACE form of the RECOVER utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - The information displayed in the RECOVER Utility (Table Spaces) panel includes one row for every matching entry found in the SYSCOPY catalog table. The match is performed on database name and table space name, and the type (**Typ**) of the entry must be either *"F"* (Full image copy) or *"I"* (Partial image copy). Only cataloged

image copy data sets are supported. This means that there can be 0, 1, or many entries for any table space name.

- If you entered information in the **Database name** or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
- To select a table space from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.
- Figure 90: Sample JCL generated for RECOVER Utility (table spaces) on page 369 shows an example of generated JCL for recovering table spaces.

Figure 90. Sample JCL generated for RECOVER Utility (table spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP RECV1: RECOVER TABLESPACE DSNDB01.SCT02
000010 //*********************************************************************
000011 //RECV1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSIN    DD *
000015 RECOVER TABLESPACE DSNDB01.SCT02    TOCOPY
000016   DFA2.IMAGCOPY.SCT02 TOVOLUME CATALOG
000017 REBUILD INDEX (ALL) TABLESPACE DSNDB01.SCT02
000018   SORTNUM 4 SORTDEVT SYSALLDA
000019 /*
****** **************************** Bottom of Data ****************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:

- ZDT/Db2 displays an alternative form of the REBUILD Utility (Indexes) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the RECOVER statement.
- Figure 91: Sample JCL generated for RECOVER Utility (table spaces) with LISTDEF specified on page 370 shows an example of generated JCL for recovering table spaces where a LISTDEF has been specified.

Figure 91. Sample JCL generated for RECOVER Utility (table spaces) with LISTDEF specified

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP RECV1: RECOVER TABLESPACE LIST LDEFNAME1
000010 //*********************************************************************
000011 //RECV1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSIN    DD *
000015 RECOVER LIST LDEFNAME1
000016 /*
****** ************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

RECOVER (index spaces) on page 370

Related references

Db2 Utilities panel on page 627
RECOVER Utility (Table Spaces) panel on page 783
RECOVER Utility (Table Spaces) with LISTDEF panel on page 785

RECOVER (index spaces)

This option corresponds to the RECOVER INDEXSPACE form of the RECOVER utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name** or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select an index space from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected index spaces.

- Figure 92: Sample JCL generated for RECOVER Utility (index spaces) on page 371 shows an example of generated JCL for recovering index spaces.

  Figure 92. Sample JCL generated for RECOVER Utility (index spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP RECV1: RECOVER INDEXSPACE JLV3D81A.XDEPT2
000010 //*********************************************************************
000011 //RECV1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSIN    DD *
000015 RECOVER INDEXSPACE JLV3D81A.XDEPT2    TOCOPY
000016   SHRIKES.JLV3D81A.XDEPT2.F00000.D2541023 TOVOLUME CATALOG
000017 /*
****** ************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:
  - ZDT/Db2 displays an alternative form of the RECOVER Utility (Index Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the RECOVER statement.

Related tasks

RECOVER (table spaces) on page 368

Related references

Db2 Utilities panel on page 627
RECOVER Utility (Index Spaces) panel on page 780
RECOVER Utility (Index Spaces) with LISTDEF panel on page 782

REORG utility

To select the REORG utility, enter `5` in the **Db2® Utility** input field on the Db2® Utilities panel.

If you selected Table spaces in the **Object Type** section of the panel, the REORG Utility (Table spaces) panel is displayed.

If you selected Indexes in the **Object Type** section of the panel, the REORG Utility (Indexes) panel is displayed.

Note: If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the REORG Utility (Table spaces) panel or REORG Utility (Indexes) panel.

Related tasks

REORG (table spaces) on page 372
REORG (indexes) on page 374

Related references

Db2 Utilities panel on page 627
REORG Utility (Table Spaces) panel on page 795
REORG Utility (Table Spaces) with LISTDEF panel on page 797
REORG Utility (Indexes) panel on page 792
REORG Utility (Indexes) with LISTDEF panel on page 793

REORG (table spaces)

To include a UNLDDN clause in the generated REORG statement, specify the required template name in **UNLDDN template name** for the REORG TABLESPACE statement on the second TEMPLATE Options panel.

Note: The **Unload External** option on the REORG Utility (Table Spaces) panel for the selected table space must be set to *"Y"*.

This option corresponds to the REORG TABLE SPACE utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name** or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select a table space from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.

◦ Figure 93: Sample JCL generated for REORG Utility (table spaces) on page 373 shows an example of generated JCL for reorganizing table spaces.

Figure 93. Sample JCL generated for REORG Utility (table spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000010 //*********************************************************************
000011 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000012 //*
000013 //*********************************************************************
000014 //* STEP RORG1: REORG TABLESPACE DSN8D81A.DSN8S81D
000015 //*********************************************************************
000016 //RORG1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000017 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000018 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000019 //DSNUPROC.SYSREC   DD UNIT=SYSALLDA,
000020 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000021 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000022 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000023 //DSNUPROC.SORTOUT  DD UNIT=SYSALLDA,
000024 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000025 //DSNUPROC.SYSIN    DD *
000026 REORG TABLESPACE DSN8D81A.DSN8S81D    LOG NO
000027   SORTDATA
000028 /*
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

• If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:

- ZDT/Db2 displays an alternative form of the REORG (Table Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the REORG statement.
- Figure 94: Sample JCL generated for REORG Utility (table spaces) with LISTDEF specified on page 374 shows an example of generated JCL for recovering table spaces where a LISTDEF has been specified.

Figure 94. Sample JCL generated for REORG Utility (table spaces) with LISTDEF specified

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000010 //*********************************************************************
000011 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000012 //*
000013 //*********************************************************************
000014 //* STEP RORG1: REORG TABLESPACE LIST LDEFNAME1
000015 //*********************************************************************
000016 //RORG1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000017 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000018 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000019 //DSNUPROC.SYSREC   DD UNIT=SYSALLDA,
000020 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000021 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000022 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000023 //DSNUPROC.SORTOUT  DD UNIT=SYSALLDA,
000024 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000025 //DSNUPROC.SYSIN    DD *
000026 REORG TABLESPACE LIST LDEFNAME1    LOG NO
000027   SORTDATA
000028 /*
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

REORG (indexes) on page 374

Setting options for Db2 utilities on page 348

Related references

Db2 Utilities panel on page 627

TEMPLATE Options (2 of 2) panel on page 898

REORG Utility (Table Spaces) panel on page 795

REORG Utility (Table Spaces) with LISTDEF panel on page 797

REORG (indexes)

This option corresponds to the REORG INDEX utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name**, **Table/index owner**, or **Table/index name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select an index from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected indexes.
  - Figure 95: Sample JCL generated for REORG Utility (indexes) on page 375 shows an example of generated JCL for reorganizing indexes.

    Figure 95. Sample JCL generated for REORG Utility (indexes)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000010 //*********************************************************************
000011 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000012 //*
000013 //*********************************************************************
000014 //* STEP RORG1: REORG INDEX
000015 //* "BUDGER2"."PLAN_TABLE_HINT_IX"
000016 //*********************************************************************
000017 //RORG1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000018 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000019 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000020 //DSNUPROC.SYSREC   DD UNIT=SYSALLDA,
000021 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000022 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000023 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000024 //DSNUPROC.SORTOUT  DD UNIT=SYSALLDA,
000025 //          SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000026 //DSNUPROC.SYSIN    DD *
000027 REORG INDEX
000028 "BUDGER2"."PLAN_TABLE_HINT_IX"
000029 /*
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:

- ZDT/Db2 displays an alternative form of the REORG (indexes) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the REORG statement.
- Figure 96: Sample JCL generated for REORG Utility (indexes) with LISTDEF specified on page 376 shows an example of generated JCL for recovering table spaces where a LISTDEF has been specified.

Figure 96. Sample JCL generated for REORG Utility (indexes) with LISTDEF specified

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000010 //*********************************************************************
000011 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000012 //*
000013 //*********************************************************************
000014 //* STEP RORG1: REORG INDEX LIST LDEF1
000015 //*********************************************************************
000016 //RORG1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000017 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000018 //         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000019 //DSNUPROC.SYSREC   DD UNIT=SYSALLDA,
000020 //         SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000021 //DSNUPROC.SYSUT1   DD UNIT=SYSALLDA,
000022 //         SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000023 //DSNUPROC.SORTOUT  DD UNIT=SYSALLDA,
000024 //         SPACE=(&ALLOC,(&PRI,&SEC),RLSE)
000025 //DSNUPROC.SYSIN    DD *
000026 REORG INDEX LIST LDEF1
000027 /*
****** *************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

REORG (table spaces) on page 372

Related references

Db2 Utilities panel on page 627
REORG Utility (Indexes) panel on page 792
REORG Utility (Indexes) with LISTDEF panel on page 793

RUNSTATS utility

To select the RUNSTATS utility, enter `6` in the **Db2® Utility** input field on the Db2® Utilities panel.

If you selected Table spaces in the **Object Type** section of the panel, the Runstats Utility (Table spaces) panel is displayed.

If you selected Indexes in the **Object Type** section of the panel, the Runstats Utility (Indexes) panel is displayed.

Note: If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the Runstats Utility (Table spaces) panel or Runstats Utility (Indexes) panel.

Related tasks

RUNSTATS (table spaces) on page 377
RUNSTATS (indexes) on page 378

Related references

Db2 Utilities panel on page 627
RUNSTATS Utility (Table Spaces) panel on page 808
RUNSTATS Utility (Table Spaces) with LISTDEF panel on page 810
RUNSTATS Utility (Indexes) panel on page 805
RUNSTATS Utility (Indexes) with LISTDEF panel on page 807

RUNSTATS (table spaces)

This option corresponds to the RUNSTATS TABLE SPACE form of the RUNSTATS utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name** or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select a table space from those displayed, type s in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.
  - Figure 97: Sample JCL generated for RUNSTATS Utility (table spaces) on page 377 shows an example of generated JCL for gathering statistics on a table space.

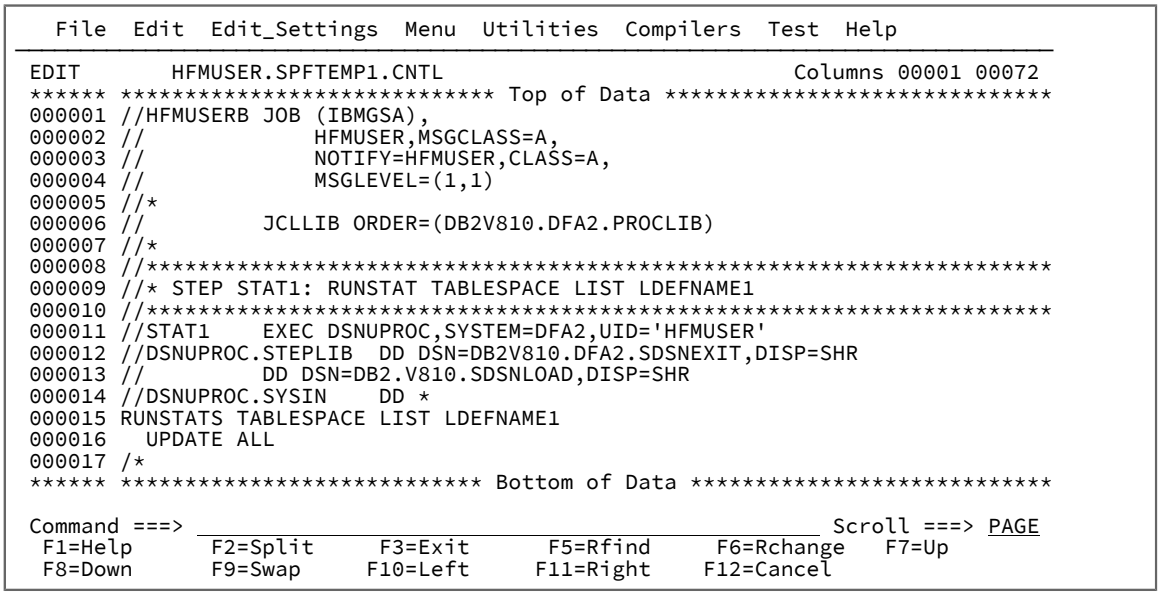
  Figure 97. Sample JCL generated for RUNSTATS Utility (table spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP STAT1: RUNSTAT TABLESPACE DSN8D81A.DSN8S81D
000010 //*********************************************************************
000011 //STAT1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSIN    DD *
000015 RUNSTATS  TABLESPACE DSN8D81A.DSN8S81D
000016   UPDATE ALL
000017 /*
****** **************************** Bottom of Data ****************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:
  - ZDT/Db2 displays an alternative form of the RUNSTATS Utility (Table Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the RUNSTATS statement.
  - Figure 98: Sample JCL generated for RUNSTATS (table spaces) with LISTDEF specified on page 378 shows an example of generated JCL for recovering table spaces where a LISTDEF has been specified.

  Figure 98. Sample JCL generated for RUNSTATS (table spaces) with LISTDEF specified



Related tasks

RUNSTATS (indexes) on page 378

Related references

Db2 Utilities panel on page 627
RUNSTATS Utility (Table Spaces) panel on page 808
RUNSTATS Utility (Table Spaces) with LISTDEF panel on page 810

RUNSTATS (indexes)

This option corresponds to the RUNSTATS INDEX form of the RUNSTATS utility control statement.

- If you did *not* specify the name of a LISTDEF on the Db2® Utilities panel:
  - If you entered information in the **Database name**, **Table/index owner**, or **Table/index space name** filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.
  - To select an index from those displayed, type `s` in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected indexes.

- Figure 99: Sample JCL generated for RUNSTATS Utility (indexes) on page 379 shows an example of generated JCL for gathering statistics on indexes.

  Figure 99. Sample JCL generated for RUNSTATS Utility (indexes)

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP STAT1: RUNSTAT INDEX
000010 //* "BUDGER2"."PLAN_TABLE_HINT_IX"
000011 //*********************************************************************
000012 //STAT1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000013 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000014 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000015 //DSNUPROC.SYSIN    DD *
000016 RUNSTATS INDEX (
000017 "BUDGER2"."PLAN_TABLE_HINT_IX"
000018    )
000019   UPDATE ALL
000020 /*
****** ***************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

- If you *did* specify the name of a LISTDEF on the Db2® Utilities panel:
  - ZDT/Db2 displays an alternative form of the RUNSTATS Utility (Indexes) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the RUNSTATS statement.
  - Figure 100: Sample JCL generated for RUNSTATS (indexes) with LISTDEF specified on page 379 shows an example of generated JCL for recovering table spaces where a LISTDEF has been specified.

    Figure 100. Sample JCL generated for RUNSTATS (indexes) with LISTDEF specified

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //          JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //* STEP STAT1: RUNSTAT INDEX LIST LDEFNAME1
000010 //*********************************************************************
000011 //STAT1    EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000012 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000013 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000014 //DSNUPROC.SYSIN    DD *
000015 RUNSTATS INDEX LIST LDEFNAME1
000016   UPDATE ALL
000017 /*
****** ***************************** Bottom of Data ***************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

RUNSTATS (table spaces) on page 377

Related references

Db2 Utilities panel on page 627
RUNSTATS Utility (Indexes) panel on page 805
RUNSTATS Utility (Indexes) with LISTDEF panel on page 807

UNLOAD utility

The UNLOAD utility unloads data from one or more source objects to one or more sequential data sets in external format. It is the preferred method for unloading or transferring large amounts of data.

You can specify the Db2® templates that identify the data set to receive the LOAD utility control statements (PUNCHDDN clause) and the data sets into which the data is unloaded (UNLDDN clause).

To select the UNLOAD utility, enter `7` in the **Db2® utility** input field on the Db2® Utilities panel.

If you selected *"Tables"* in the **Object Type** section of the Db2® Utilities panel, the UNLOAD Utility (Tables) panel is displayed.

If you selected *"Table spaces"* in the **Object Type** section of the Db2® Utilities panel, the UNLOAD Utility (Table Spaces) panel is displayed.

Note: If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the UNLOAD Utility (Tables) panel or UNLOAD Utility (Table Spaces) panel.

Related tasks

UNLOAD (tables) on page 381
UNLOAD (table spaces) on page 386
Setting options for Db2 utilities on page 348

Related references

Db2 Utilities panel on page 627
TEMPLATE Options (2 of 2) panel on page 898
UNLOAD Utility (Tables) "From" panel on page 910
UNLOAD Utility (Tables) "To" panel on page 911
UNLOAD Utility (Tables) with LISTDEF panel on page 914
UNLOAD Utility (Table Spaces) panel on page 908
UNLOAD Utility (Table Spaces) with LISTDEF panel on page 909

UNLOAD (tables)

Use the UNLOAD (tables) utility to unload data into a sequential data set. You can choose to replace any existing data with the new data, or to add the new data to the data already in the data set. The UNLOAD (tables) utility provides the most efficient means of adding large amounts of data to a data set.

When you use the UNLOAD utility, you can:

- Select the columns to be unloaded by specifying the required columns in the *"From"* template
- Reformat data during the unload by mapping table columns to fields in the output data set (as defined by a template)
- Create and save a Z Data Tools Base component template that describes the unloaded data
- Select which rows to unload
- Specify the number of rows to be unloaded

To include a PUNCHDDN clause in the generated UNLOAD statement, specify the required template name in **PUNCHDDN template name** on the second TEMPLATE Options panel.

To include a UNLDDN clause in the generated UNLOAD statement, specify the required template name in **UNLDDN template name** for the UNLOAD statement on the second TEMPLATE Options panel.

The following description applies if you did *not* specify the name of a LISTDEF on the Db2® Utilities panel. If you *did* specify the name of a LISTDEF on the Db2® Utilities panel, see UNLOAD (tables) with LISTDEF specified on page 386.

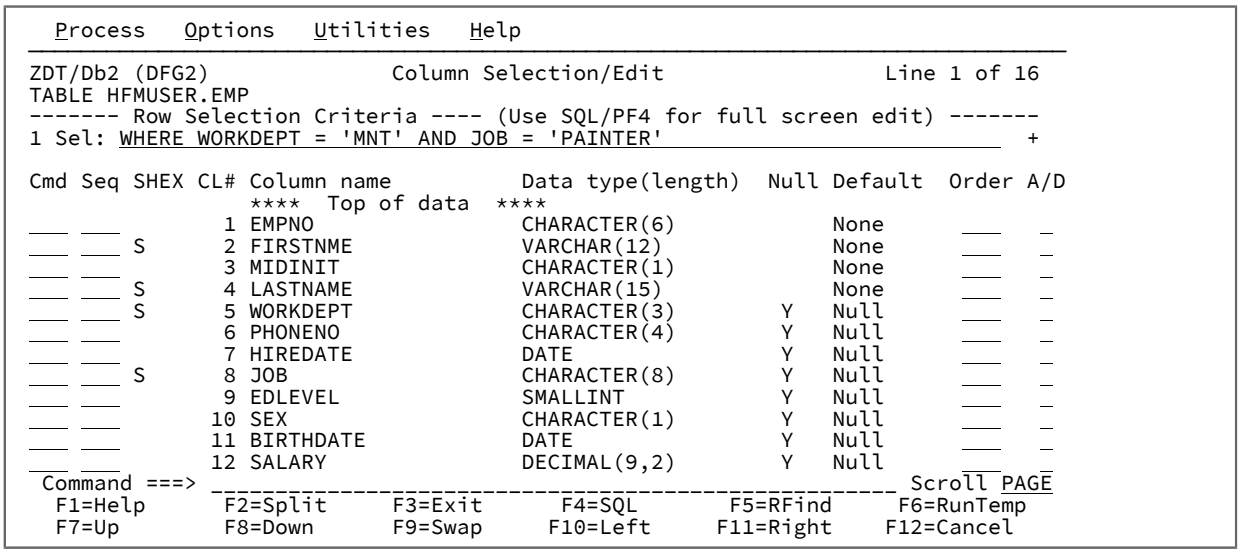
On the UNLOAD Utility (Tables) panel:

1. The table name details you entered on the Db2® Utilities panel are copied to the UNLOAD Utility (Tables) panel. If you do not want to unload all of the rows in the source table, specify the number of rows you want to unload in the **Unload Count** field.
2. If you are using a specific template for the source table, specify the name of the template in the **From Template** entry fields.
3. In the **Processing Options**, specify the template you want ZDT/Db2 to use for the source table, and whether you want to edit the template before ZDT/Db2 builds the utility batch job.
4. To proceed, press Enter.

   If you selected the **Edit template** option, ZDT/Db2 displays the Column Selection/Edit panel. Use this panel to restrict the data in the source table by:
   - Selecting or deselecting specific columns for unloading
   - Specifying row selection criteria. ZDT/Db2 selects only the rows whose contents satisfy the criteria for unloading

   For example, the template shown in Figure 101: Column Selection/Edit panel showing row selection criteria and selected columns on page 382 has been edited to select certain columns and only rows whose contents meet certain criteria. If you use this template as the *"From"* template for an unload, the data unloaded is restricted to:
   - The columns FIRSTNME, LASTNAME, WORKDEPT, and JOB, but only where WORKDEPT equals *"MNT"* and JOB equals *"PAINTER"*.

Figure 101. Column Selection/Edit panel showing row selection criteria and selected columns



If you did not select the **Edit template** option (or have completed the edit of the template option), and the value of the **UNLDDN template name** field on the TEMPLATE Options (2 of 2) panel is blank, ZDT/Db2 displays the UNLOAD Utility *"To"* panel.

Note: If the **UNLDDN template name** field is not blank, ZDT/Db2 does not display the UNLOAD Utility *"To"* panel as the output is already defined.

5. To temporarily change the settings for the UNLOAD options that ZDT/Db2 uses in building the current batch job, select **Edit UNLOAD utility options** to display the UNLOAD Options panel.

   If you change any of the original settings, they remain in effect only for as long as you use the Db2® Utilities UNLOAD feature in which the settings were changed. If you exit from the UNLOAD Utility (Tables) panel and return to this panel later, the settings changed in the previous UNLOAD session are unavailable.

6. On the UNLOAD Utility *"To"* panel, specify the sequential data set that is the target of the unload by entering the *"To"* data set details using the **Data set name** and, optionally, **Volume**, entry fields.
7. If you are using a specific template or copybook for the *"To"* data set, you can specify the name of the template or copybook data set and, optionally, the member name, in the **To Data Template or Copybook** entry fields.

   The data format used by the Db2® UNLOAD utility is fixed. Therefore, if you specify a copybook or template data set, the copybook or template should reflect the Db2® UNLOAD data format to avoid data mismatch problems.

   When a template is specified for the Db2® unload data set, it must be a Z Data Tools Base component template; a ZDT/Db2 template cannot be used.

   When the **Allow padding** option on the Db2® UNLOAD utility options panel is not selected, ZDT/Db2 cannot generate a template that describes the unloaded data. In this situation, all fields related to template processing are protected when the UNLOAD utility *"To"* panel is displayed and the template usage option is set to 5 (NONE).

8. Select the **Template usage** and **Disposition** options you want.

9. You can review the Db2® UNLOAD options by selecting the **View UNLOAD options** field. No changes can be made when the Db2® UNLOAD utility options panel is displayed. To make changes return to the Db2® UNLOAD utility *"From"* panel and select **Edit UNLOAD options**.
10. To edit the template mapping for the source table, select **Edit template mapping** to display the Template Mapping panel.
11. To proceed, press Enter.

    ZDT/Db2 builds a batch job to execute the Db2® UNLOAD utility job using the specified input details.

Figure 102: Sample JCL generated for UNLOAD (Tables) on page 384 shows an example of the generated JCL for unloading a table object.

Figure 102. Sample JCL generated for UNLOAD (Tables)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //*
000010 //* ZDT/Db2 GENERATED JOB TO UNLOAD A TABLE
000011 //*
000012 //*********************************************************************
000013 //*
000014 //*********************************************************************
000015 //* STEP UNLOAD: UNLOAD FROM TABLE
000016 //*   "HFMUSER"."EMP"
000017 //*********************************************************************
000018 //UNLOAD1  EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000019 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000020 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000021 //DSNUPROC.SYSREC00 DD DISP=(MOD,CATLG,DELETE),UNIT=SYSALLDA,
000022 //          BLKSIZE=0,
000023 //          AVGREC=U,
000024 //          LRECL=82,RECFM=FB,DSORG=PS,
000025 //          SPACE=(82,(42,4),RLSE),
000026 //          DSN=HFMUSER.FRED
000027 //DSNUPROC.SYSPUNCH DD DISP=(NEW,CATLG,DELETE), UTIL CONTRL DATA
000028 //          UNIT=SYSALLDA,SPACE=(TRK,(1,1),RLSE),
000029 //          DSN=HFMUSER.HFMUSER.EMP.D2581536.SYSPUNCH
000030 //DSNUPROC.SYSIN    DD *
000031 UNLOAD DATA FROM TABLE
000032 "HFMUSER"."EMP"
000033   HEADER NONE
000034   LIMIT ALL
000035   (
000036    "FIRSTNME          " POSITION(1     ) VARCHAR
000037   ,"LASTNAME          " POSITION(15    ) VARCHAR
000038   ,"WORKDEPT          " POSITION(33    ) CHAR(3)
000039   ,"JOB               " POSITION(38    ) CHAR(8)
000040   ,"BONUS             " POSITION(48    ) DECIMAL
000041   ,"COMM              " POSITION(66    ) DECIMAL
000042   )
000043   WHEN
000044   (
000045   "WORKDEPT" = 'MAINT' AND "JOB" = 'PAINTER'
000046   )
000047   UNLDDN(SYSREC00)
000048   NOSUBS
000049 /*
****** ************************** Bottom of Data ***************************

Command ===>                                                 Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Working with templates on page 74

Editing a template on page 84

Letting ZDT/Db2 do the mapping on page 113

Setting options for Db2 utilities on page 348

Related references

Column Selection/Edit panel on page 497
UNLOAD Utility (Tables) "From" panel on page 910
UNLOAD Utility (Tables) "To" panel on page 911
TEMPLATE Options (2 of 2) panel on page 898
UNLOAD Options panel on page 905
Template Mapping panel on page 900
TEMPLATE Options (2 of 2) panel on page 898

Null indicator byte processing

The Db2® UNLOAD utility data format uses a single byte for the Db2® null value, located immediately before the data for each nullable column, in the output data set. The value in this byte is set to X'00' if the column contains a value and X'FF' when the column contains the Db2® null value. The Db2® UNLOAD utility automatically places the appropriate value in the null indicator byte in the output data set.

ZDT/Db2 automatically generates a Z Data Tools Base component template when the template processing option is set to 3 or 4 on the *"To"* panel. This template includes fields for the null indicator bytes as appropriate.

When a copybook is used to describe the output data, ZDT/Db2 attempts to generate a reasonable mapping between the template for the Db2® object and the template for the output data set. The data format used by the Db2® UNLOAD utility is fixed, therefore the data format described by the copybook should reflect this format, to avoid data mismatch problems.

The Db2® null indicator byte in a user-specified copybook is successfully mapped when:

- It immediately precedes a nonvarying target field, or precedes the length field of a varying target field.
- The null indicator byte does not overlap any field that has already been mapped.
- The field name for the null indicator byte contains "NULL". In this case, ZDT/Db2 documents the mapping as comments in the generated JCL for the Db2® UNLOAD utility.

Other types of mapping, or null indicator byte usage are not supported, including:

- A copybook that would map a nullable Db2® column to a data field without a null indicator byte.
- A copybook that would map a not-nullable Db2® column to a data field with a null indicator byte.
- A copybook that uses null indicator fields that are not one byte in length, or are not located immediately before the data for the column to which they pertain.

Related references

UNLOAD Options panel on page 905

UNLOAD (tables) with LISTDEF specified

If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the UNLOAD Utility (Tables) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the UNLOAD statement.

Figure 103: Sample JCL generated for UNLOAD (Tables) with LISTDEF specified on page 386 shows an example of generated JCL for unloading a table object where a LISTDEF has been specified.

Figure 103. Sample JCL generated for UNLOAD (Tables) with LISTDEF specified

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //*
000010 //* ZDT/Db2 GENERATED JOB TO UNLOAD TABLE LIST
000011 //*
000012 //*********************************************************************
000013 //*
000014 //*********************************************************************
000015 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000016 //*********************************************************************
000017 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000018 //*
000019 //*********************************************************************
000020 //* STEP UNLOAD1: UNLOAD TABLE LIST LDEFNAME1
000021 //*********************************************************************
000022 //UNLOAD1  EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000023 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000024 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000025 //DSNUPROC.SYSREC00 DD DISP=(NEW,CATLG,DELETE),
000026 //          UNIT=,SPACE=(&ALLOC,(&PRI,&SEC),RLSE),
000027 //          DSN=HFMUSER.LDEFNAME.D2601243.SYSREC
000028 //DSNUPROC.SYSPUNCH DD DISP=(NEW,CATLG,DELETE), UTIL CONTRL DATA
000029 //          UNIT=,SPACE=(TRK,(1,1),RLSE),
000030 //          DSN=HFMUSER.LDEFNAME.D2601243.SYSPUNCH
000031 //DSNUPROC.SYSIN    DD *
000032 UNLOAD LIST LDEFNAME1
000033   UNLDDN(SYSREC00)
000034   NOSUBS
000035 /*
****** **************************** Bottom of Data ****************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related references

UNLOAD Utility (Tables) with LISTDEF panel on page 914

UNLOAD (table spaces)

This option corresponds to the UNLOAD TABLESPACE form of the UNLOAD utility control statement.

When you use UNLOAD (table spaces), you can use the following combinations of filter fields on the Db2® Utilities panel:

- **Database name**, or **Table/index space name**.

If you entered information in any of the preceding filter fields on the Db2® Utilities panel, the displayed rows are limited to the Db2® objects that match the information you specified.

Figure 104: UNLOAD Utility (Table Spaces) panel showing list of table spaces. on page 387 shows an example of the UNLOAD Utility (Table Spaces) panel with a list of table spaces.

To select an index space from those displayed, type s in the **Sel** column. An asterisk (*) in the **PRC** column indicates selected table spaces.

Figure 104. UNLOAD Utility (Table Spaces) panel showing list of table spaces.

```
  Process   Options   Utilities   Help
_______________________________________________________________________________
ZDT/Db2 (DFG2)          UNLOAD Utility (Table Spaces)           Row 1 to 4 of 4

S P          Table      Part'n
e R Database Space      Range   Number Number       Allocation
l C Name     Name       One Two Part'n Tables     Primary Secdry       Space

S   BILLDB1  ACT        2__ 3__      1      1           3      3          48
_   BILLDB1  ACT        ___ ___      2      1           5      5         144
_   BILLDB1  ACT        ___ ___      3      1           3      3         144
_   BILLDB1  ACT        ___ ___      4      1           5      5          48
****************************** Bottom of data ********************************




Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

If only a single row results from the filter values you entered, the row is marked as selected, and is shown with an asterisk (*) in the **PRC** field.

If you entered a table space name containing wildcards, the ISPF part of the UNLOAD Utility panel is initialized with one row for each of the table space names that match the input specification. Note that if you did not specify a database name on the Db2® utilities panel, or specified a generic name, the list may contain table spaces from more than one database.

To select a table spaces to copy, type an *"S"* in the **SEL** field. Selected table spaces have an asterisk (*) in the **PRC** field.

You can modify the UNLOAD utility statement by changing the value in the **Part'n Range** input field.

In Figure 104: UNLOAD Utility (Table Spaces) panel showing list of table spaces. on page 387, the input shows that a range of partitions (from 2 to 3) is to be unloaded.

Figure 105: Sample JCL generated for UNLOAD Utility (Table Spaces) on page 388 shows an example of generated JCL for unloading a table object where a LISTDEF has been specified.

Figure 105. Sample JCL generated for UNLOAD Utility (Table Spaces)

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //*
000010 //* ZDT/Db2 GENERATED JOB TO UNLOAD TABLESPACE
000011 //*
000012 //*********************************************************************
000013 //*
000014 //*********************************************************************
000015 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000016 //*********************************************************************
000017 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000018 //*
000019 //*********************************************************************
000020 //* STEP UNLOAD1: UNLOAD TABLESPACE DSN8D81A.DSN8S81D
000021 //*********************************************************************
000022 //UNLOAD1  EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000023 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000024 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000025 //DSNUPROC.SYSREC00 DD DISP=(NEW,CATLG,DELETE),
000026 //          UNIT=SYSALLDA,SPACE=(&ALLOC,(&PRI,&SEC),RLSE),
000027 //          DSN=HFMUSER.DSN8D81A.DSN8S81D.D2581554.SYSREC
000028 //DSNUPROC.SYSPUNCH DD DISP=(NEW,CATLG,DELETE), UTIL CONTRL DATA
000029 //          UNIT=SYSALLDA,SPACE=(TRK,(1,1),RLSE),
000030 //          DSN=HFMUSER.DSN8D81A.DSN8S81D.D2581554.SYSPUNCH
000031 //DSNUPROC.SYSIN    DD *
000032 UNLOAD TABLESPACE DSN8D81A.DSN8S81D
000033   UNLDDN(SYSREC00)
000034   NOSUBS
000035 /*
****** **************************** Bottom of Data ****************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related references

Column Selection/Edit panel on page 497
UNLOAD Utility (Table Spaces) panel on page 908
TEMPLATE Options (2 of 2) panel on page 898
UNLOAD Options panel on page 905
Template Mapping panel on page 900

UNLOAD (table spaces) with LISTDEF specified

If you specified the name of a LISTDEF on the Db2® Utilities panel, ZDT/Db2 displays an alternative form of the UNLOAD Utility (Table Spaces) panel showing the specified LISTDEF and a number of editable fields that correspond to keywords in the UNLOAD statement.

Figure 106: Sample JCL generated for UNLOAD (Table Space) with LISTDEF specified on page 389 shows an example of generated JCL for unloading a table space object where a LISTDEF has been specified.

Figure 106. Sample JCL generated for UNLOAD (Table Space) with LISTDEF specified

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       HFMUSER.SPFTEMP1.CNTL                          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //HFMUSERB JOB (IBMGSA),
000002 //             HFMUSER,MSGCLASS=A,
000003 //             NOTIFY=HFMUSER,CLASS=A,
000004 //             MSGLEVEL=(1,1)
000005 //*
000006 //         JCLLIB ORDER=(DB2V810.DFA2.PROCLIB)
000007 //*
000008 //*********************************************************************
000009 //*
000010 //* ZDT/Db2 GENERATED JOB TO UNLOAD TABLESPACE LIST
000011 //*
000012 //*********************************************************************
000013 //*
000014 //*********************************************************************
000015 //* STEP SETPARM: OVERRIDE ANY GLOBAL DATASET SIZING VALUES HERE
000016 //*********************************************************************
000017 //SETPARM SET ALLOC='TRK',PRI='1',SEC='1'
000018 //*
000019 //*********************************************************************
000020 //* STEP UNLOAD1: UNLOAD TABLESPACE LIST LDEF1
000021 //*********************************************************************
000022 //UNLOAD1  EXEC DSNUPROC,SYSTEM=DFA2,UID='HFMUSER'
000023 //DSNUPROC.STEPLIB  DD DSN=DB2V810.DFA2.SDSNEXIT,DISP=SHR
000024 //          DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
000025 //DSNUPROC.SYSREC00 DD DISP=(NEW,CATLG,DELETE),
000026 //          UNIT=SYSALLDA,SPACE=(&ALLOC,(&PRI,&SEC),RLSE),
000027 //          DSN=HFMUSER.LDEF1.D2581557.SYSREC
000028 //DSNUPROC.SYSPUNCH DD DISP=(NEW,CATLG,DELETE), UTIL CONTRL DATA
000029 //          UNIT=SYSALLDA,SPACE=(TRK,(1,1),RLSE),
000030 //          DSN=HFMUSER.LDEF1.D2581557.SYSPUNCH
000031 //DSNUPROC.SYSIN    DD *
000032 UNLOAD TABLESPACE LIST LDEF1
000033   UNLDDN(SYSREC00)
000034   NOSUBS
000035 /*
****** **************************** Bottom of Data ****************************

Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related references

UNLOAD Utility (Table Spaces) with LISTDEF panel on page 909

Chapter 13. Working with SQL statements

When you want to:

- Issue any dynamic SQL statement from your panel or from a data set, or
- Build and run an SQL SELECT statement interactively

you can use the SQL prototyping, execution and analysis option (4).

Note:

1. This chapter does not attempt to describe the syntax of SQL statements. For full details, see the *DB2 for z/OS SQL Reference*.

To display the SQL Prototyping, Execution and Analysis panel, from the Primary Option Menu panel select option 4 (SQL).

Related tasks

Should you use basic or advanced SQL prototyping? on page 390
Using basic SQL prototyping on page 390
Using advanced SQL prototyping on page 399

Related references

Primary Option Menu panel on page 760
SQL Prototyping, Execution and Analysis panel on page 843

Should you use basic or advanced SQL prototyping?

If you only need to build a simple SQL SELECT statement, basic SELECT prototyping 4.1) is probably sufficient for your needs. If you need to build a more complex statement, use advanced SELECT prototyping (4.2).

Related references

Using basic SQL prototyping on page 390
Using advanced SQL prototyping on page 399

Using basic SQL prototyping

Basic SELECT prototyping uses two panels: you use the primary panel to specify the table (or tables) that contain the data you want to retrieve (FROM clause), and the secondary panel to specify the columns you want to show, any qualifying information (WHERE clause), and the sequence in which you want the data to be shown (ORDER BY clause).

To display the primary Basic SELECT Prototyping panel, select option 1 (Basic) on the SQL Prototyping, Execution and Analysis panel.

Figure 107. Primary Basic SELECT Prototyping panel

```
Process   Options   Utilities   Help
 ZDT/Db2 (DFG2)             Basic SELECT Prototyping

 Enter the name(s) of the table(s) from which to retrieve data:
     Owner           Name
   1 DSN81010      + EMP                          + Location ________________
   2 DSN81010      + DEPT                         + Database ________
   3 ____________  + ____________________________  + Tbl spc. ________
   4 ____________  + ____________________________  +
   5 ____________  + ____________________________  +
   6 ____________  + ____________________________  +
   7 ____________  + ____________________________  +
   8 ____________  + ____________________________  +
   9 ____________  + ____________________________  +
  10 ____________  + ____________________________  +
  11 ____________  + ____________________________  +
  12 ____________  + ____________________________  +
  13 ____________  + ____________________________  +
  14 ____________  + ____________________________  +
  15 ____________  + ____________________________  +
 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F9=Swap      F12=Cancel
```

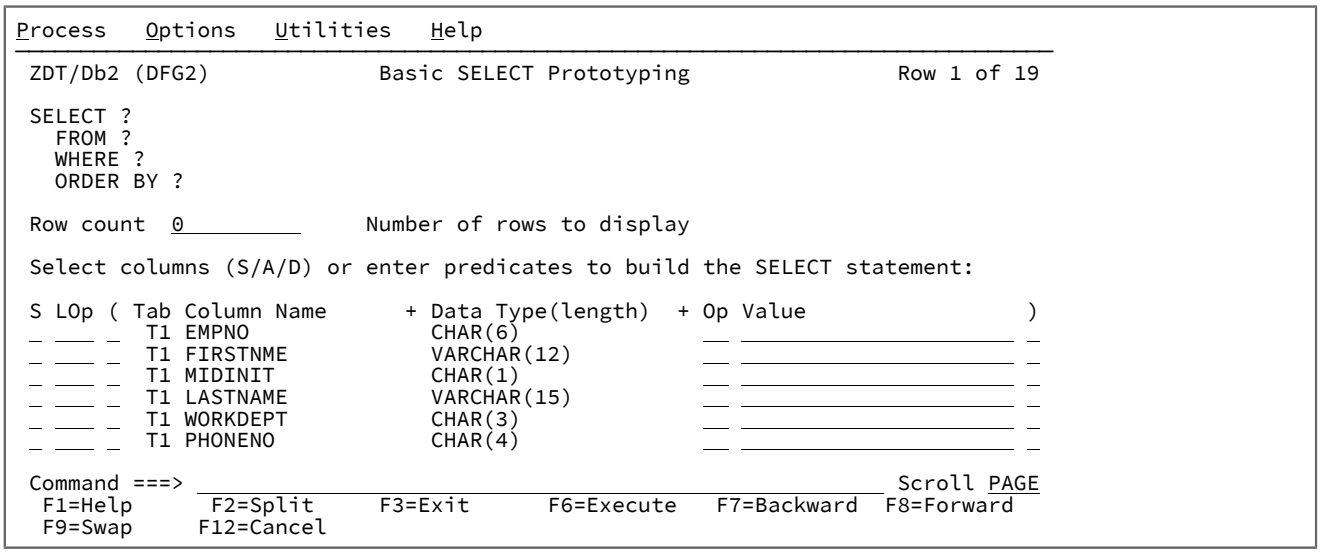
On this panel you can specify up to 15 tables, views, synonyms, or aliases that can be part of the SELECT statement being prototyped. If you specify more than one object, the columns for all objects specified are shown on the secondary Basic SELECT Prototyping panel when you press Enter; see Figure 108: Secondary Basic SELECT Prototyping panel on page 392.

You cannot modify the entry-line numbers at the left of the panel (1, 2, and so on). The purpose of the entry-line number is to provide a suffix for the correlation name used to identify the Db2® object. Db2® correlation names are discussed in detail in the *SQL Reference*. The complete Db2® correlation name is shown in the "Tab" column on the secondary Basic SELECT Prototyping panel when you press Enter. The complete correlation name will be a capital letter, followed by the entry-line number. The capital letter will usually be 'T', but may be another letter if any of the Db2® object names start with 'T'. The Basic Prototyper ensures that the letter selected for the correlation name reference is not the same as the first letter of any Db2® object name included on the primary Basic SELECT Prototyping panel.

The columns and rows included in the result table are determined by the SELECT and WHERE clauses specified on the secondary Basic SELECT Prototyping panel, which is displayed when you specify the objects and press Enter. The examples in this section use the tables DSN81010.EMP and DSN81010.DEPT.

Tables and views are specified by optionally entering values in the **Location**, **Database**, **Tbl spc.** (table space), and **Owner** fields, and entering values in the **Name fields**. Synonyms are specified by entering values in the **Name fields**. If a value is entered in the **Owner** field for a synonym, it must be the current SQLID. All of the fields support catalog searches by specifying a pattern in the field. After you have specified the table names on the primary Basic SELECT Prototyping panel, press Enter to display the secondary Basic SELECT Prototyping panel.

Figure 108. Secondary Basic SELECT Prototyping panel



When initially displayed, the secondary Basic SELECT Prototyping panel shows all of the columns for the Db2® objects entered on the primary panel. You can scroll up or down if there are more columns than will fit on the display. The SQL statement being prototyped is shown at the top of the panel as 4 clauses (SELECT, FROM, WHERE and ORDER BY). An empty clause is indicated by a "*?*".

ZDT/Db2 automatically updates one or more of these clauses as you enter either primary or line commands. Each of the clause fields is an ISPF expandable field. If the field is too short to display all of the clause then a "***+***" will be shown at the end of the field. You can position the cursor anywhere in the field and press the ISPF EXPAND key to show the complete clause in a pop-up window.

To limit the number of rows returned in the result set when the SQL statement is executed, specify the maximum number of rows in the **Row count** field.

The following primary commands are available when you use basic SELECT prototyping:

ALL

Use this command to select all columns for a Db2® object. You can specify an optional parameter *#n* to include only the columns of the *n*th Db2® object displayed.

CANCEL

Use this command to return to the primary Basic SELECT Prototyping panel

EXECUTE

Use this command to run the SQL statement being prototyped

RESET

Use this command to abandon all changes to the SQL statement. All clauses are returned to empty status.

SQL

Use this command to show the complete SQL statement in an ISPF editor session

UNDO

Use this command to back-out the last change made.

The following line commands can be entered against any column:

- S or /

  You type either S or / against the entry for a column to add that column to the list of columns for the SQL statement. Data from the column will be included in the result table when the statement is executed.

  You can select multiple columns at once, however your entries are processed from top to bottom, so you might need to select columns individually if the order of columns in the result table is important.

- A and D

  You type either A or D against the entry for a column to add an ORDER BY clause for that column. If the specified column does not exist in the column list, the column is also added to the column list. Data from the column will be included in the result table when the statement is executed, and rows will by sorted in ascending or descending order on that data.

  Entering a A results in the ORDER BY clause including the column name without modification, this defaults to ascending (ASC keyword).

  Entering a D results in the ORDER BY clause including the column name followed by the DESC keyword, this indicates the result table should be sorted in descending order based on the data for the column.

- U

  You type U against the entry for a column to remove the last column reference from the list of columns in the result table. When the column is removed and the column is no longer specified in the list of columns:

  - All ORDER BY clause references to the column are also removed.
  - All predicate references to the column are also removed.

  If the column has been added to the column list multiple times, use the U command repeatedly to remove all references.

- UO

  You type UO against the entry for a column to remove the last ORDER BY reference.

  Removing an ORDER BY reference for the column does not remove the column from the column list, nor does it remove any predicates referencing the column from the WHERE clause.

  If the column has been added to the ORDER BY clause multiple times, use the UO command repeatedly to remove all references.

- UW

  You type UW against the entry for a column to remove the last predicate that references the column.

Removing a WHERE clause reference for the column does not remove the column from the column list, nor does it remove any references for the column from the ORDER BY clause.

If there are multiple predicates containing the column, use the UW command repeatedly to remove all references.

Specifying a predicate

You specify a predicate (for example `EMPNO = '000010'`) by typing the comparison operator (*"="*) in the Op field, and the comparison value (*"000010"*) in the Value field. To specify a join predicate, include the correlation reference for the table containing the join column in the value field.

For example, to specify a join between DSN81010.EMP.WORKDEPT and DSN81010.DEPT.DEPTNO:

- Type *"="* in the **Op** field for column `WORKDEPT`.
- Type *"T2.DEPTNO"* in the **Value** column for `WORKDEPT`.

Figure 109: Specifying a join predicate on page 394 shows an example of specifying the join, and Figure 110: The result of the join predicate shown in the previous figure on page 395 shows the result of the join predicate.

Figure 109. Specifying a join predicate

```
Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
 ZDT/Db2 (DFG2)             Basic SELECT Prototyping               Row 1 of 19

 SELECT T1.*
   FROM "DSN81010"."EMP" T1
   WHERE ?
   ORDER BY ?

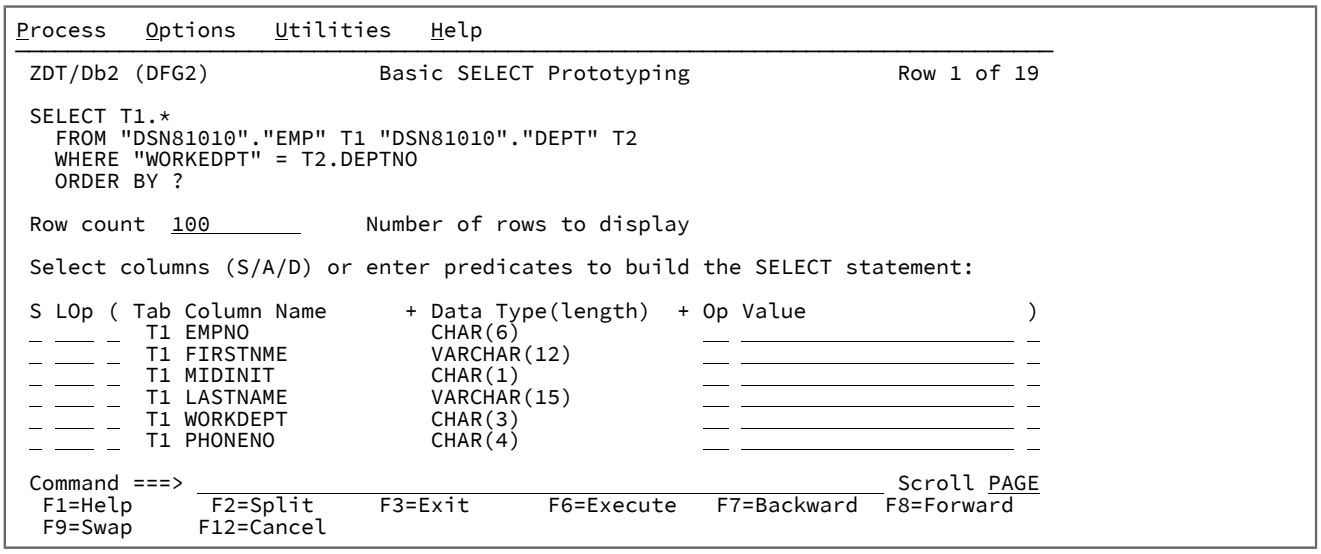
 Row count  100          Number of rows to display

 Select columns (S/A/D) or enter predicates to build the SELECT statement:

 S LOp ( Tab Column Name      + Data Type(length)  + Op Value                 )
 _ ___ _  T1 EMPNO              CHAR(6)              __ ____________________ _
 _ ___ _  T1 FIRSTNME           VARCHAR(12)          __ ____________________ _
 _ ___ _  T1 MIDINIT            CHAR(1)              __ ____________________ _
 _ ___ _  T1 LASTNAME           VARCHAR(15)          __ ____________________ _
 _ ___ _  T1 WORKDEPT           CHAR(3)              =  T2.DEPTNO           _
 _ ___ _  T1 PHONENO            CHAR(4)              __ ____________________ _

 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F6=Execute   F7=Backward  F8=Forward
  F9=Swap     F12=Cancel
```

Figure 110. The result of the join predicate shown in the previous figure



Example of using basic SELECT prototyping

We'll use an example to show how to use basic SELECT prototyping to build a SELECT statement and then run it to see the results.

Suppose you want to list details of all employees in the table DSN8810.EMP whose:

- Department is not *"E01"*, and
- JOB is *"Manager" or* SALARY is at least $40,000

For each employee, you want to show the following columns from the table DSN8810.EMP (correlation name #1):

LASTNAME
FIRSTNME
WORKDEPT
JOB
SALARY (in ascending sequence)

and the following column from the table DSN8810.EEMP (correlation name #2):

BONUS

You can specify details on the secondary Basic SELECT Prototyping panel for more than one column at a time. However, for the purpose of clarity, the following steps show the details being entered one at a time for each column, and the resultant SELECT statement that ZDT/Db2 progressively builds (and displays at the top of the panel) after you press Enter.

Step 1

Select the column, LASTNAME, by typing `s` in the **S** entry field:

```
S LOp ( Tab Column Name        Data Type(length)    Op Value                 )
⋮
s ___ _  #1 LASTNAME           VARCHAR(15)             _____________________ _
⋮
```

ZDT/Db2 updates the generated SELECT statement by:

- Adding the name of the selected column (LASTNAME) to the SELECT clause.

```
SELECT #1.LASTNAME
  FROM DSN8810.EMP #1, DSN8810.EEMP #2
  WHERE ?
  ORDER BY ?
```

Step 2

Select the column, FIRSTNME, by typing `s` in the **S** entry field:

```
S LOp ( Tab Column Name        Data Type(length)    Op Value                 )
⋮
s ___ _  #1 FIRSTNME           VARCHAR(12)             _____________________ _
⋮
```

ZDT/Db2 updates the generated SELECT statement by:

- Adding the name of the selected column (FIRSTNME) to the SELECT clause.

```
SELECT #1.LASTNAME, #1.FIRSTNME
  FROM DSN8810.EMP #1, DSN8810.EEMP #2
  WHERE ?                          STNAME <> 'SMITH'
  ORDER BY ?
```

Step 3

To exclude employees in WORKDEPT "E01":

- Select the column, WORKDEPT, by typing `s` in the **S** entry field,
- Type the expression operator `<>` in the **Op** entry field, and
- Type `E01` in the **Value** entry field:

```
S LOp ( Tab Column Name        Data Type(length)    Op Value                 )
⋮
s ___ _  #1 WORKDEPT           CHAR(3)              <> e01_________________ _
⋮
```

ZDT/Db2 updates the generated SELECT statement by:

- Adding the name of the selected column (WORKDEPT) to the SELECT clause.
- Adding a predicate, `#1.WORKDEPT <> 'E01'`, to the WHERE clause.

```
SELECT #1.LASTNAME, #1.FIRSTNME, #1.WORKDEPT
  FROM DSN8810.EMP #1, DSN8810.EEMP #2
  WHERE #1.WORKDEPT <> 'E01'
  ORDER BY ?
```

Step 4

To select employees with a JOB of “Manager”:

- Select the column, JOB, by typing `S` in the **S** entry field,
- Type an opening parenthesis, `(`, in the **(** entry field, and
- Type `MANAGER` in the **Value** entry field:

```
S LOp ( Tab Column Name        Data Type(length)    Op Value                 )
⋮
s ___ (  #1 JOB                CHAR(8)                 manager_____________ _
⋮
```

ZDT/Db2 updates the generated SELECT statement by:

- Adding the name of the selected column (JOB) to the SELECT clause.
- Inserting the default logical operator `AND` before adding a predicate, `(#1.JOB = 'MANAGER'`, to the WHERE clause.

```
SELECT #1.LASTNAME, #1.FIRSTNME, #1.WORKDEPT, #1.JOB
  FROM DSN8810.EMP #1, DSN8810.EEMP #2
  WHERE #1.WORKDEPT <> 'E01' AND (#1.JOB = 'MANAGER'
  ORDER BY ?
```

Step 5

To also select employees with a SALARY of at least $40,000 (regardless of their JOB), and to specify that the results are to be shown in ascending sequence of SALARY:

- Select the column, SALARY, for ascending sequence by typing `A` in the **S** entry field,
- Type the logical operator, `OR`, in the **LOp** entry field,
- Type the expression operator `>=` in the **Op** entry field, and
- Type `40000` in the **Value** entry field.
- Type a closing parenthesis in the **)** entry field:

```
S LOp ( Tab Column Name        Data Type(length)    Op Value                 )
⋮
a or_ _  #1 SALARY             DECIMAL(9,2)         >= 40000_____________ )
⋮
```

ZDT/Db2 updates the generated SELECT statement by:

- Adding the name of the selected column (SALARY) to the SELECT clause.
- Adding a predicate, `OR #1.SALARY >= 40000)`, to the WHERE clause.
- Adding the name of the selected column (SALARY), and the keyword `ASC`, to the ORDER BY clause.

```
SELECT #1.LASTNAME, #1.FIRSTNME, #1.WORKDEPT, #1.JOB, #1.SALARY
  FROM DSN8810.EMP #1, DSN8810.EEMP #2
  WHERE #1.WORKDEPT <> 'E01' AND (#1.JOB = 'MANAGER' OR #1.SALARY >= 40000)
  ORDER BY #1.SALARY ASC
```

Step 6

Select the column, BONUS (from the table DSN8810.EEMP), by typing `s` in the **S** entry field:

```
S LOp ( Tab Column Name        Data Type(length)    Op Value                 )
⋮
s ___ _   #2 BONUS             DECIMAL(9,2)           ____________________ _
⋮
```

ZDT/Db2 updates the generated SELECT statement by:

- Adding the name of the selected column (BONUS) to the SELECT clause.

```
SELECT #1.LASTNAME, #1.FIRSTNME, #1.WORKDEPT, #1.JOB, #1.SALARY, #2.BONUS
  FROM DSN8810.EMP #1, DSN8810.EEMP #2
  WHERE #1.WORKDEPT <> 'E01' AND (#1.JOB = 'MANAGER' OR #1.SALARY >= 40000)
  ORDER BY #1.SALARY ASC
```

Figure 111: Details entered together on the secondary Basic SELECT Prototyping panel on page 398 shows how the Basic SELECT Prototyping panel is displayed if you had entered the details for all of the columns at the same time.

Note:

The effect of entering details for one column at a time (as in the previous steps) can be different to when you enter details for more than one column at a time.

For example, if you enter the details as shown in Figure 111: Details entered together on the secondary Basic SELECT Prototyping panel on page 398, FIRSTNME is shown before LASTNAME in the generated SELECT statement.

Figure 111. Details entered together on the secondary Basic SELECT Prototyping panel

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
 ZDT/Db2 (DFG2)             Basic SELECT Prototyping          Row 1 to 8 of 14

 SELECT ?
   FROM DSN8810.EMP #1, DSN8810.EEMP #2
   WHERE ?
   ORDER BY ?

 Row count  ALL_______      Number of rows to display

 Select columns (S/A/D) or enter predicates to build the SELECT statement:

 S LOp ( Tab Column Name      + Data Type(length)  + Op Value                 )
⋮
 s ___ _   #1 FIRSTNME          VARCHAR(12)             ____________________ _
⋮
 s ___ (   #1 JOB               CHAR(8)                 manager______________ _
 s ___ _   #1 LASTNAME          VARCHAR(15)             ____________________ _
⋮
 a or_ _   #1 SALARY            DECIMAL(9,2)         >= 40000_____________ )
⋮
 s ___ _   #1 WORKDEPT          CHAR(3)              <> e01_______________ _
⋮
 s ___ _   #2 BONUS             DECIMAL(9,2)            ____________________ _
⋮

 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=Expand    F6=Execute   F7=Backward
  F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Related tasks

Executing a SELECT statement after you have built it using prototyping on page 424

Related references

Basic SELECT Prototyping panel on page 476
SQL primary command on page 1005

Using advanced SQL prototyping

In this section, the following terms are used:

clause

A distinct part of the SELECT statement, such as a WHERE clause.

element

A single item within the syntax (such as a keyword, variable, reference to a fragment, or special character).

fragment

A portion of the SELECT statement syntax. Fragments can contain references to other fragments.

instance

A repeatable item that has been assigned a value. For example, the following GROUP BY clause contains three instances:

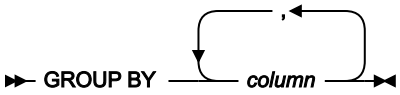
```
GROUP BY COL1 , COL2 , COL3
```

keyword

An item that must be spelled exactly as shown. Keywords (for example, `FROM`) are shown in uppercase in the syntax diagram displayed on the Advanced SELECT Prototyping panel.

repeatable item

An item that can be repeated within the syntax. Repeatable items are indicated by a repeat arrow that can contain a repeat separator. For example:



variable

An element that requires you to enter a value.

Advanced SELECT prototyping lets you develop a complex SELECT statement using a recursive set of specification panels.

Each panel displays a fragment of the SELECT statement syntax, presented in the form of a syntax diagram (in a similar way to a reference manual). You build the SELECT statement by selecting the optional fragments of the syntax you want to use, discarding those that you do not need, and entering items where required. The panels guide you through the syntax and ensure that you include all the required fragments of the syntax. Each panel shows the partly-formed SELECT statement as you progressively build it.

As you build the SELECT statement, you can return to any part of the statement to:

- Change values.
- Add a fragment of syntax.
- Remove a fragment of syntax.

Note:

1. If you have a 24-line screen, to see the SQL statement as it is being built you must remove the function key lines from your display by entering `PFSHOW OFF`.
2. If the generated SQL statement extends over several lines, you can scroll forward or backward with the Forward function key (F8) and Backward function key (F7) respectively.

To display the Advanced SELECT Prototyping panel, select option 2 (Advanced) on the SQL Prototyping, Execution and Analysis panel.

The following primary commands are available when you use advanced SELECT prototyping:

- CANCEL
- DESELECT
- EXECUTE
- INSRPT
- NEXTRPT
- PREVRPT
- SQL

Related tasks

Example 2 (Changing the SQL statement) on page 407
Example 3 (Adding to the SQL statement) on page 408
Example 4 (Removing part of the SQL statement) on page 414
Executing a SELECT statement after you have built it using prototyping on page 424

Related references

SQL Prototyping, Execution and Analysis panel on page 843
Advanced SELECT Prototyping panel on page 451
CANCEL primary command on page 926
DESELECT primary command on page 938
EXECUTE primary command on page 948
INSRPT primary command on page 971
NEXTRPT primary command on page 984

PREVRPT primary command on page 988
SQL primary command on page 1005

Tabbing and color-coding

Advanced SELECT prototyping uses colors to help you:

- Navigate the SELECT statement syntax.
- Identify current and non-current parts of the generated SELECT statement, particularly for fragments that contain repeatable items.

Syntax diagram color-coding

To help you navigate the syntax, advanced SELECT prototyping uses a tabbing and color-coding scheme.

You can tab to those parts of the syntax where you are able to either make a choice or enter a variable. You cannot tab to a required keyword.

Advanced SELECT prototyping uses the following system of colors for the syntax diagram:

RED

References to required fragments that have not been completed, or references to optional fragments that have been selected but not completed.

WHITE

Required keywords. You cannot select these, or tab to them.

GREEN

In general terms, indicates a reference to a completed fragment of the syntax:

- Selected optional keywords (otherwise yellow)
- Required fragments whose subselections have been made to enable them to be resolved (otherwise red)
- Optional fragments whose subselections have been made to enable them to be resolved (otherwise yellow)

YELLOW

Optional keywords and fragments that are not currently selected (green when selected and complete; red when selected and incomplete).

Generated SELECT statement color-coding

To help you recognize which part of the generated SELECT statement is current (the part that is accessible to you) and, in the case of repeatable items, which is the current repeatable item, advanced SELECT prototyping uses the following system of colors for the generated SELECT statement:

GREEN

The current part of the SELECT statement and which, therefore, is accessible to you. This includes any data that is part of a current repetition.

YELLOW

Part of a repetition that is not current. To make a non-current (YELLOW) current (GREEN), use the PrvRpt function key (F10) or NxtRpt function key (F11) function keys.

WHITE

Parts of the SELECT statement that are above the level of the current fragment.

Related references

Advanced SELECT Prototyping panel on page 451

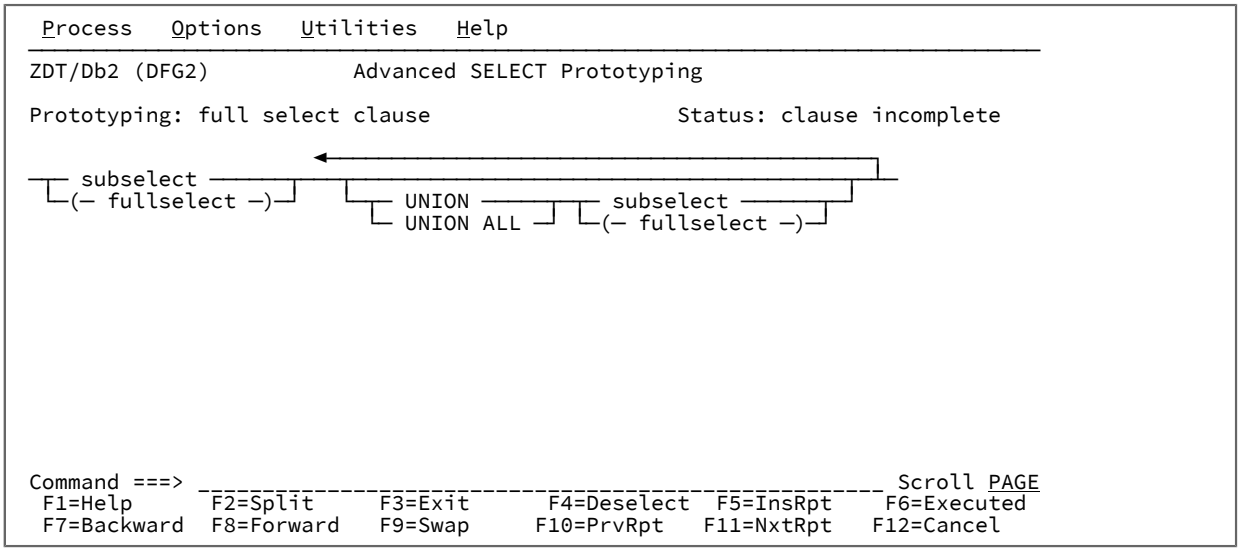
Using the advanced SELECT prototyping panels to build a SELECT statement

If you use advanced SELECT prototyping, you place the cursor on the part of the syntax you want and press Enter. To specify a value for a variable, you type the name in a panel. As you proceed to build the SELECT statement, ZDT/Db2 shows the partly-built statement within the panel you are viewing. You repeat this process until you have built the complete SELECT statement.

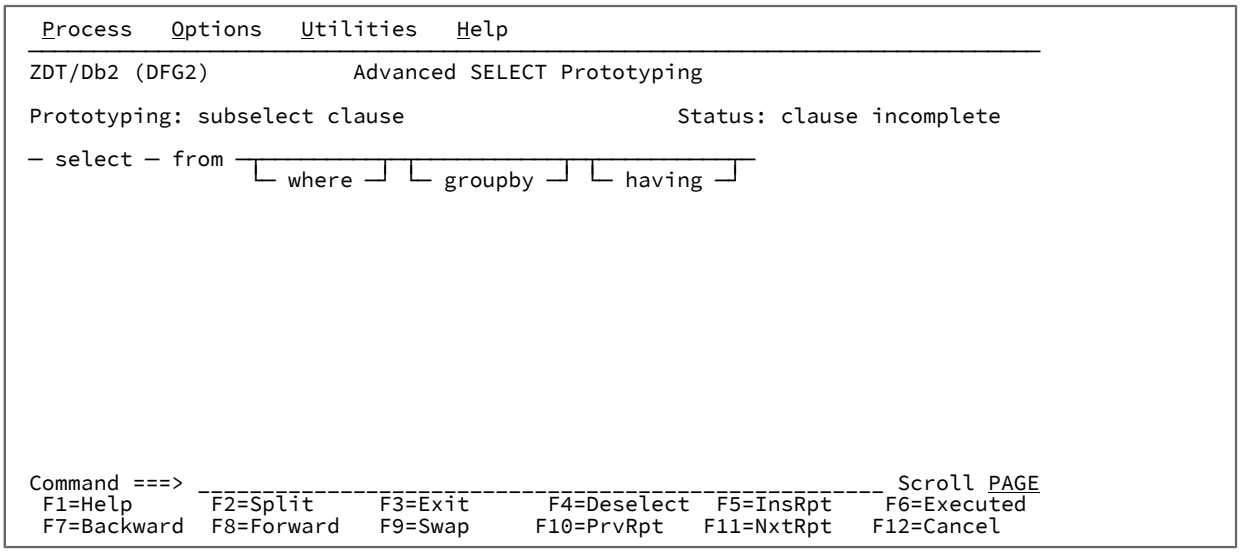
Example 1 (Building a simple SQL statement)

The following series of panels show each stage of using advanced SELECT prototyping to build an SQL statement to show all rows of the table PARTS.

1. Select option 2 (Advanced) from the SQL Prototyping and Execution panel to display the initial Advanced SELECT Prototyping panel.
2. Move the cursor to the **fullselect** fragment.
3. Press Enter. The syntax for the full SELECT clause is displayed.

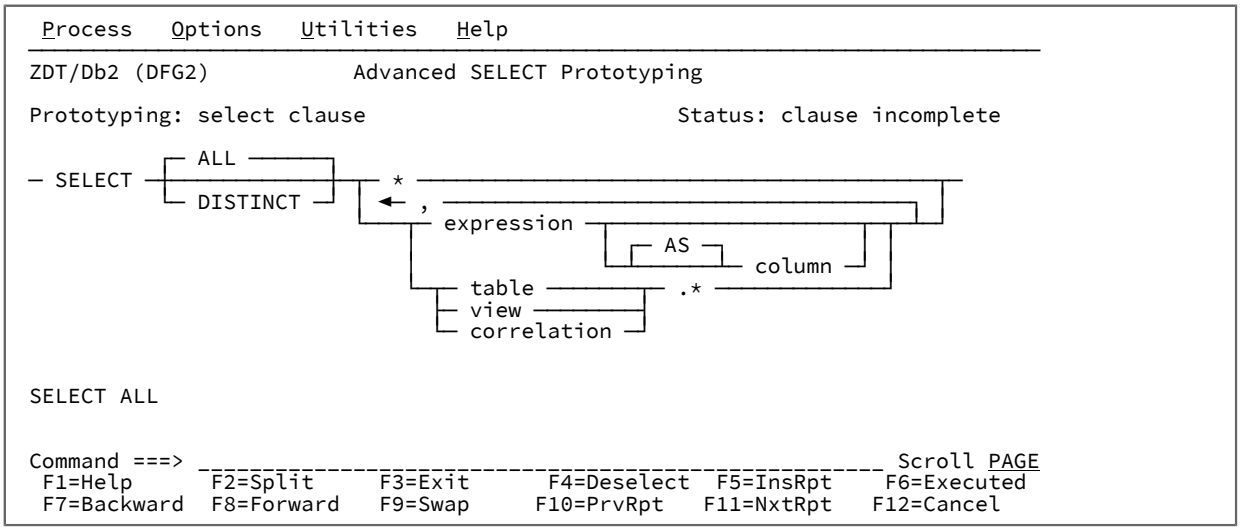


4. Move the cursor to the **subselect** fragment.
5. Press Enter. The syntax for the subselect clause is displayed.



Notice that the **select** and **from** fragments are displayed on your screen in *red* indicating that the subsections relating to these fragments must be completed before these fragments can be resolved.

6. Move the cursor to the **select** fragment.
7. Press Enter. The syntax for the SELECT clause is displayed.



The panel now shows the partly-built SELECT statement:

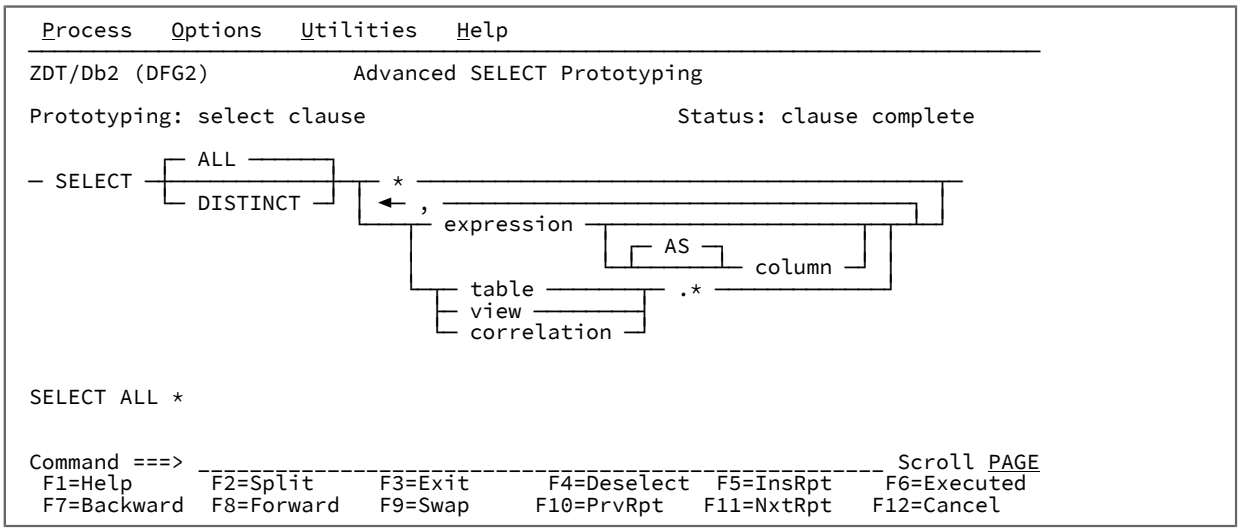
```
SELECT ALL
```

This is because:

- The keyword SELECT is a required keyword for this fragment. As you have selected this fragment, ZDT/Db2 has entered it into the SELECT statement.
- ALL is a default keyword (and, in this case, what you want).

8. Move the cursor to the asterisk (*).

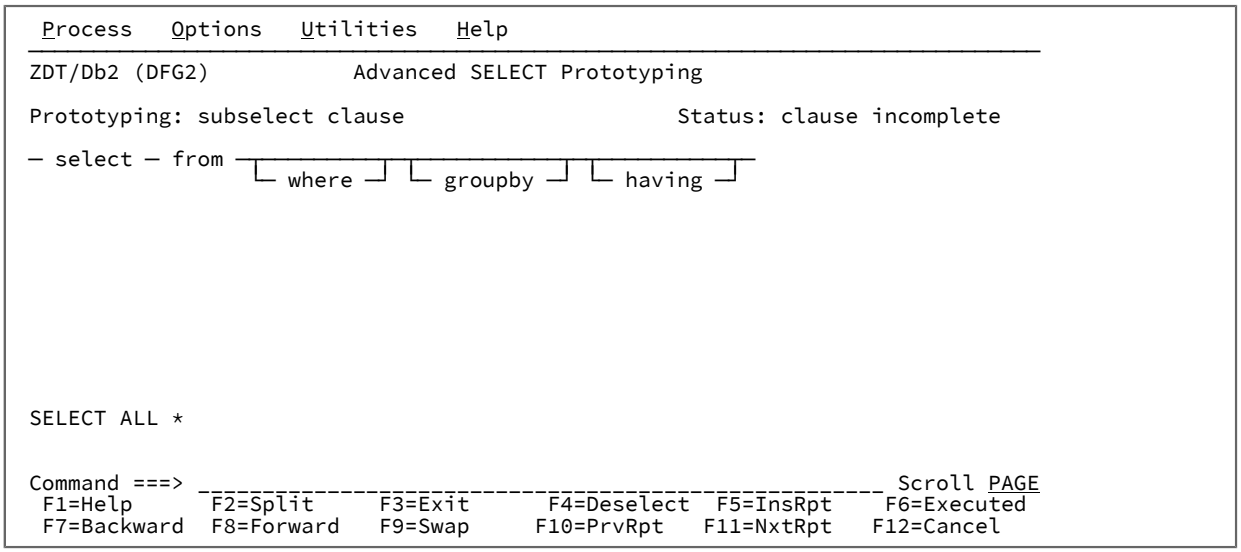
9. Press Enter.



The panel now shows the partly-built SELECT statement:

```
SELECT ALL *
```

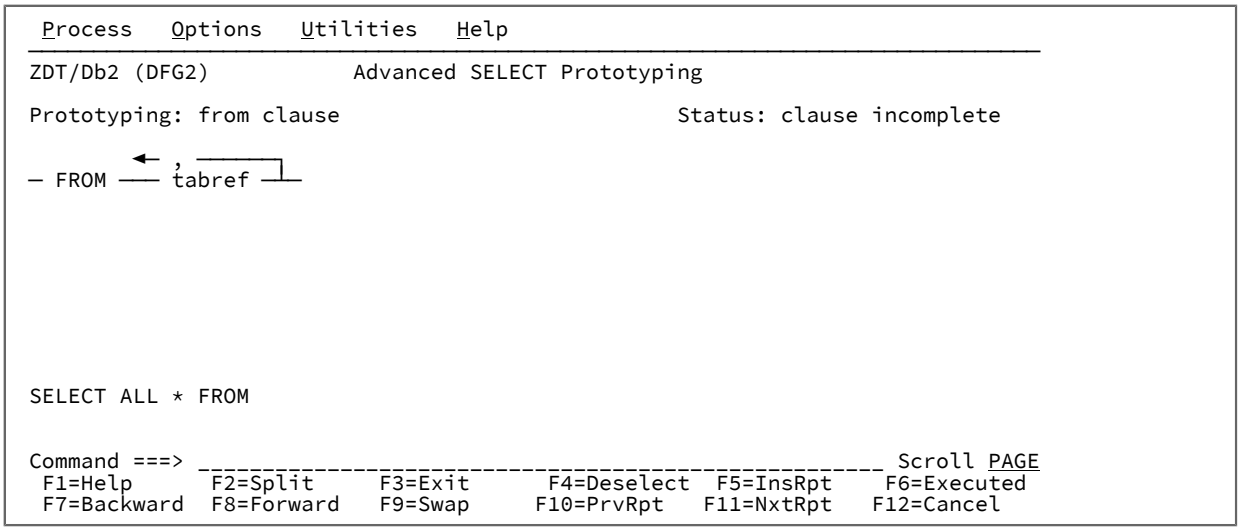
10. Press the Exit function key (F3). The syntax for the subselect clause is displayed.



Notice that the **select** fragment has now changed on your screen from red to *green* indicating that the subsection relating to this fragment has been completed and this fragment can be resolved.

You now need to complete the subsection relating to the **from** fragment.

11. Move the cursor to the **from** fragment.
12. Press Enter. The syntax for the FROM clause is displayed.



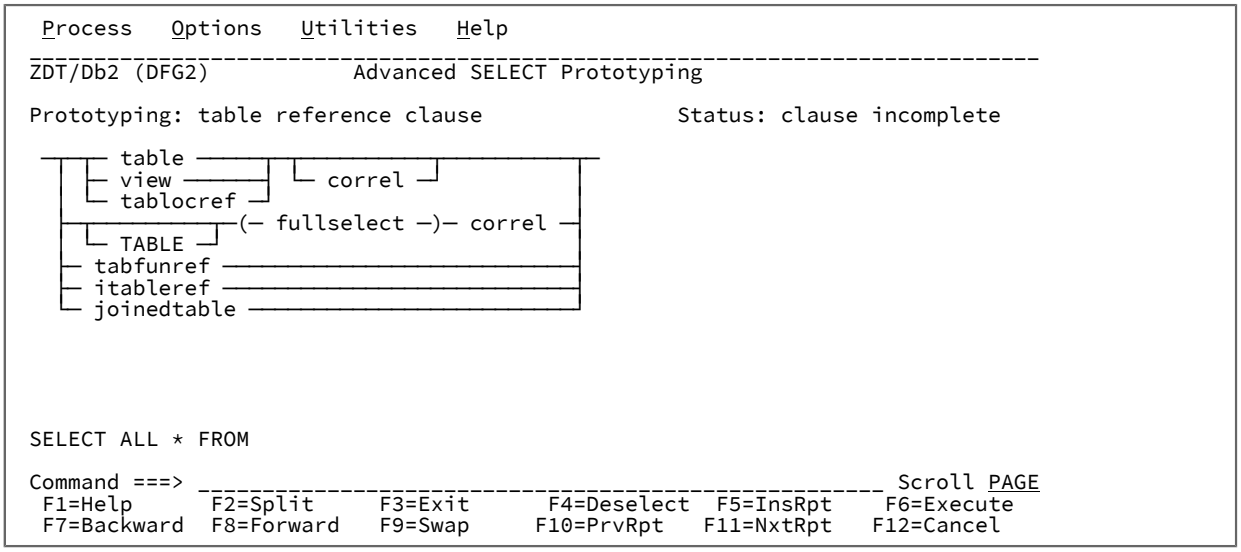
The panel now shows the partly-built SELECT statement:

```
SELECT ALL * FROM
```

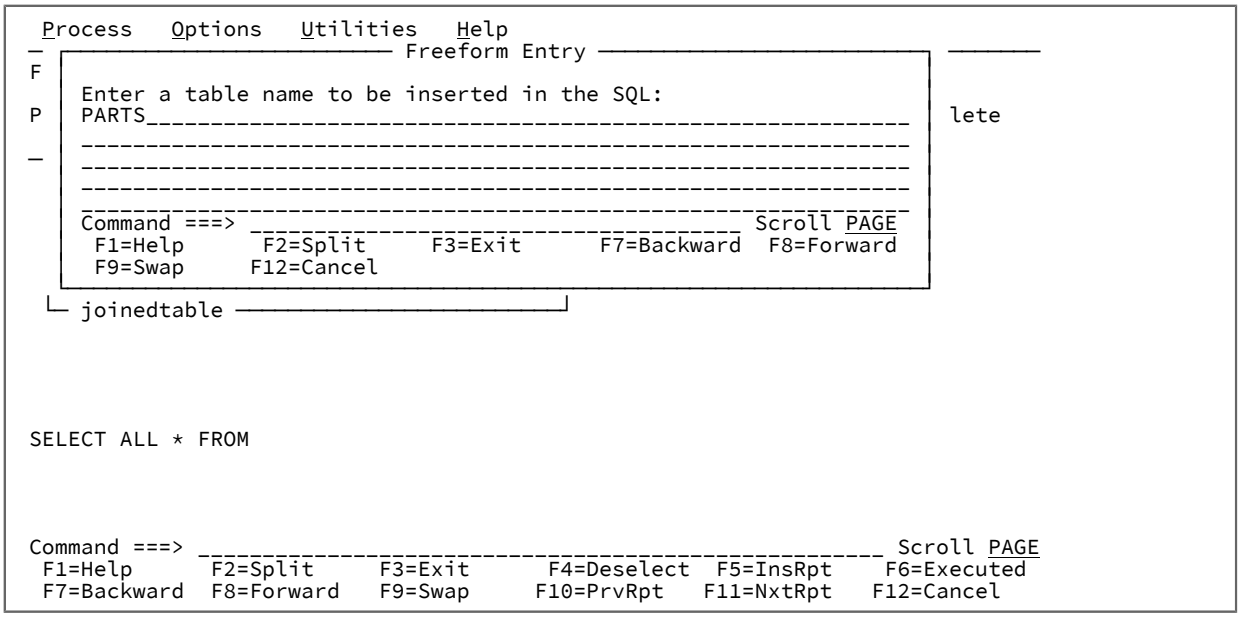
This is because the keyword FROM is a required keyword.

Again, notice that the **tabref** fragment is displayed on your screen in *red*, indicating that the subsection relating to this fragment is yet to completed before this fragment can be resolved

13. Move the cursor to the **tabref** fragment.
14. Press Enter. The syntax for the table specification clause is displayed.

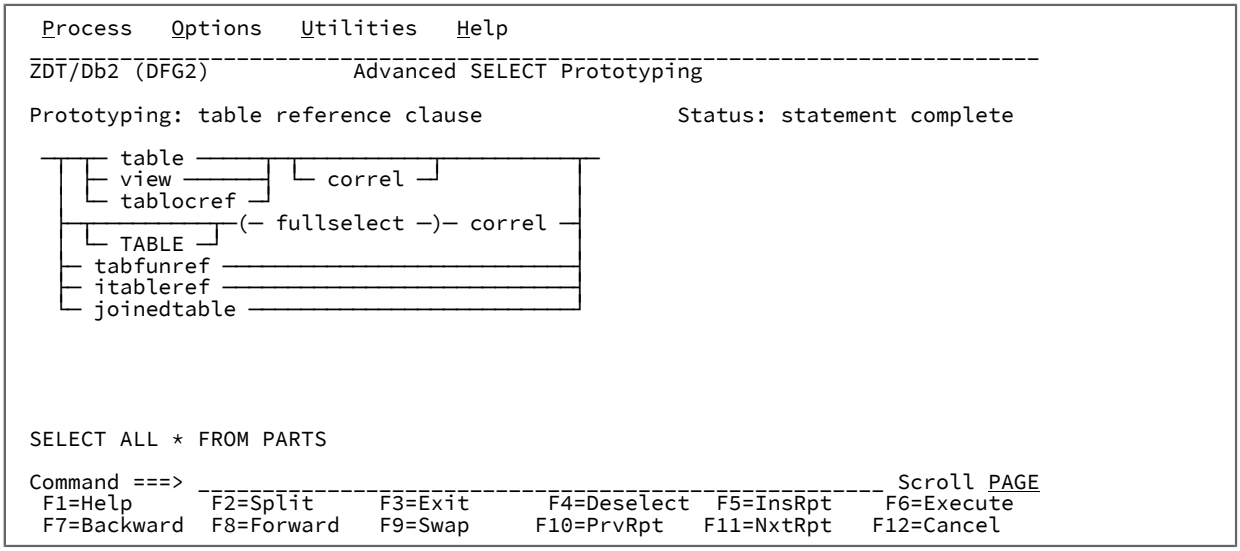


15. Move the cursor to the **table** fragment.
16. Press Enter. A panel is displayed in which you can enter a table name.
17. Type `PARTS` in the panel.



18. Press the Exit function key (F3). The syntax for the table specification clause is displayed.

Figure 112. Advanced SELECT prototyping: the completed statement (example 1)



The panel shows the completed SELECT statement:

```
SELECT ALL * FROM PARTS
```

Notice that the **table** fragment has now changed on your screen from red to *green*, indicating that the subsection relating to this fragment has been completed and the fragment can be resolved. Also notice that the status indicator (**Status:**) shows "statement complete".

Related tasks

Executing a SELECT statement after you have built it using prototyping on page 424

Related references

SQL Prototyping, Execution and Analysis panel on page 843
Advanced SELECT Prototyping panel on page 451

Changing the SQL statement

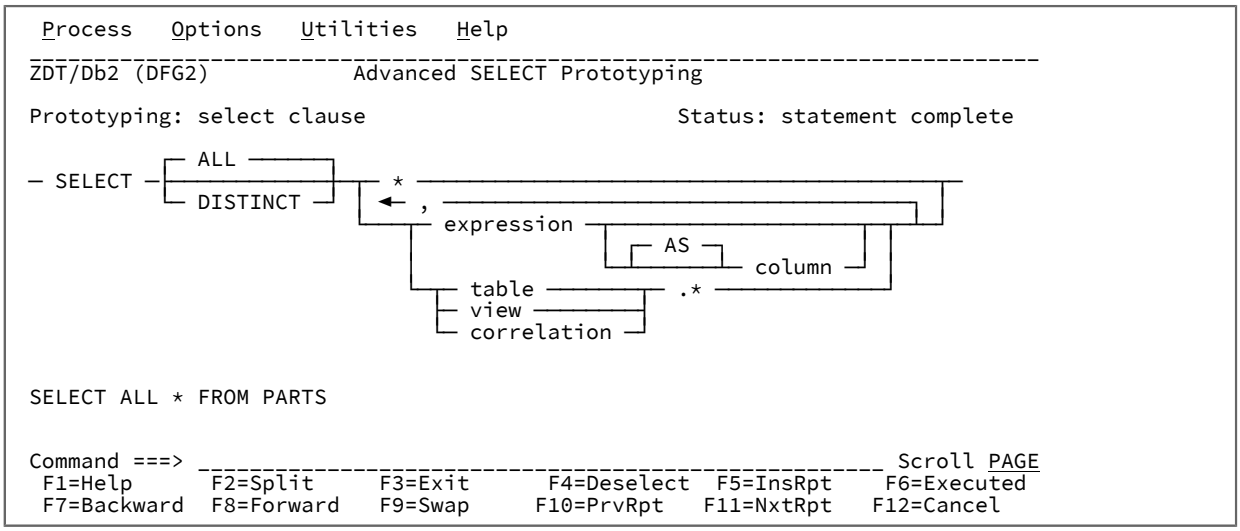
At any stage of building an SQL statement, you can:

- Change part of the statement by positioning the cursor on the relevant part of the SELECT statement and pressing Enter. ZDT/Db2 returns you to that part of the syntax. You can then make the required change.
- Add an extra fragment to the statement by positioning the cursor on the relevant part of the SELECT statement and pressing Enter. ZDT/Db2 returns you to that part of the syntax. You can then add the required fragment.
- Remove a part of the statement by pressing the Deselect function key (F4).

Example 2 (Changing the SQL statement)

Having built the SQL statement in the previous exercise to show *all* rows of the table PARTS (see Figure 112: Advanced SELECT prototyping: the completed statement (example 1) on page 406), suppose you want to change this to eliminate all but one of each set of duplicate rows:

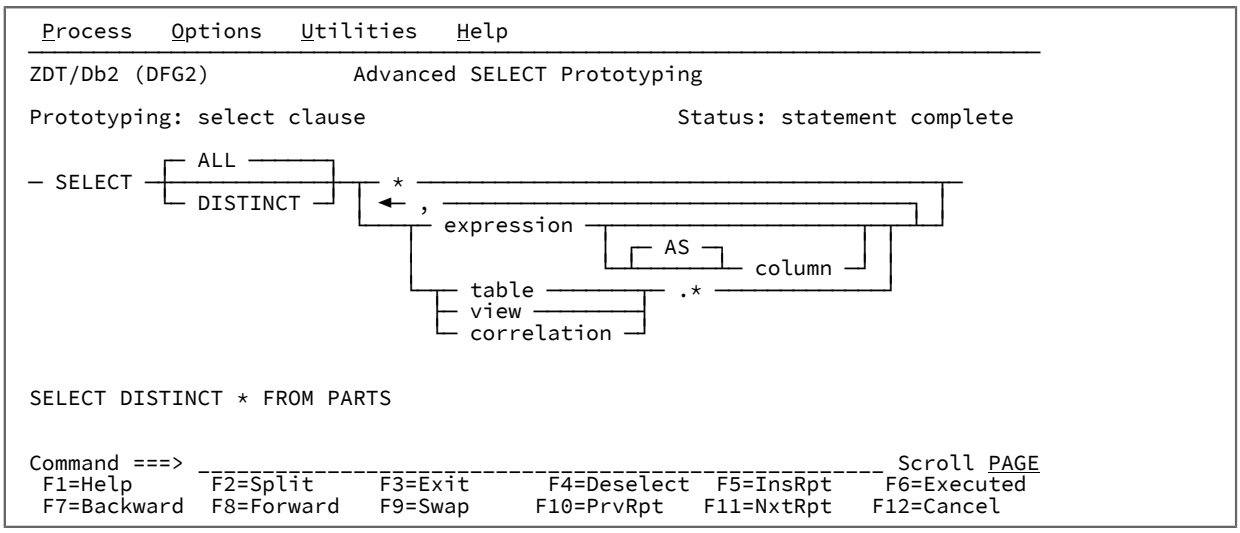
1. Move the cursor under the word ALL in the generated statement.
2. Press Enter. The syntax for the SELECT clause is displayed.



3. Move the cursor to the keyword **DISTINCT**.

4. Press Enter. DISTINCT replaces ALL in the built SELECT statement.

   Figure 113. Advanced: the completed statement (example 2)



The panel shows the completed SELECT statement:

```
SELECT DISTINCT * FROM PARTS
```

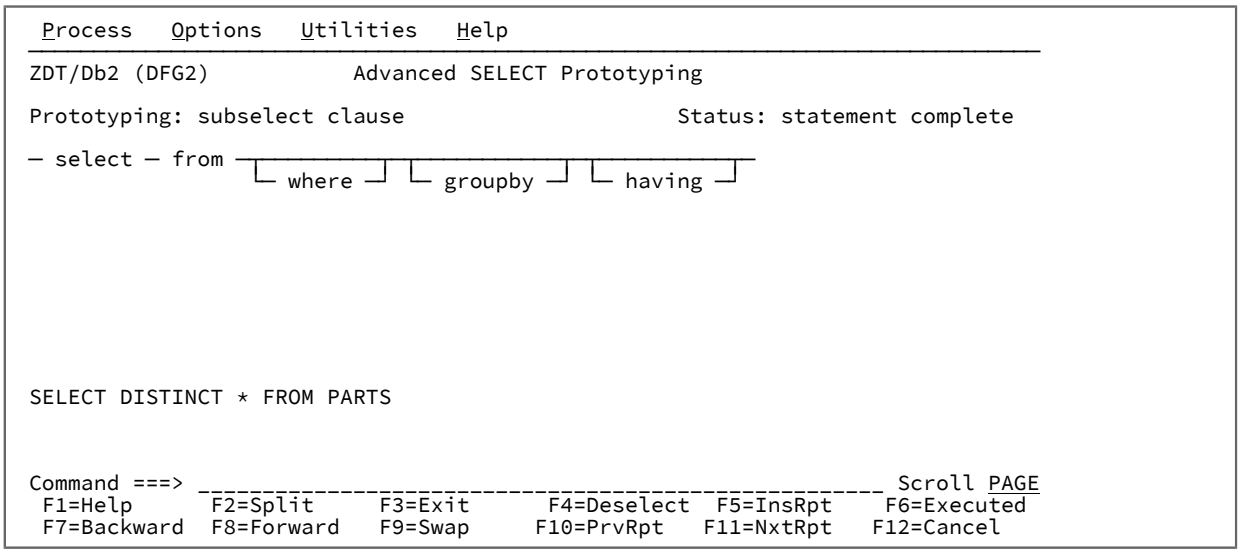
Related tasks

Example 1 (Building a simple SQL statement) on page 402
Executing a SELECT statement after you have built it using prototyping on page 424

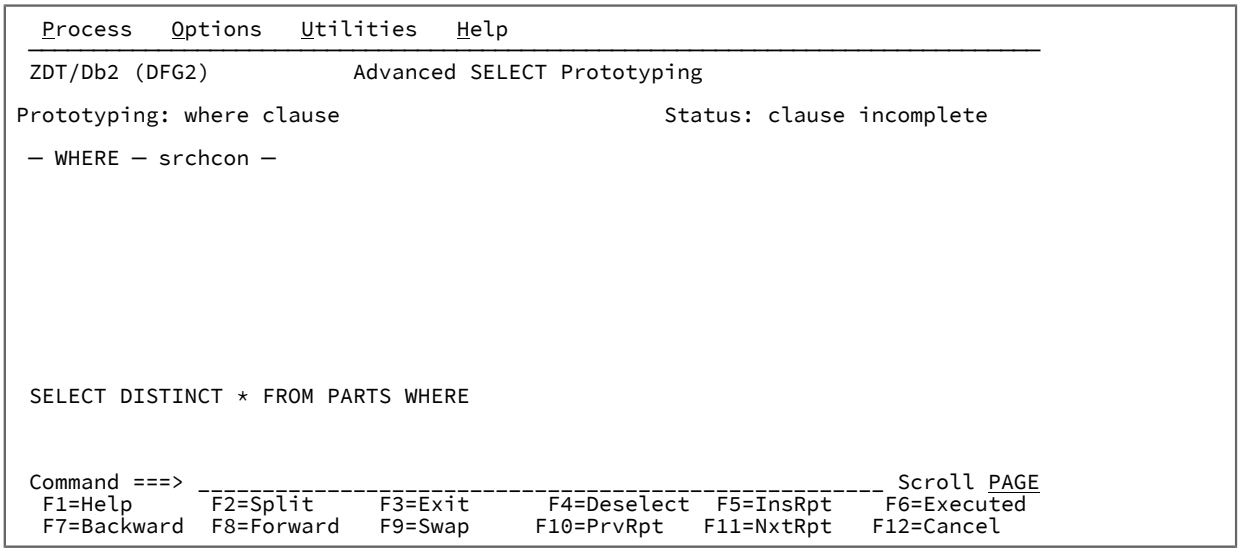
Example 3 (Adding to the SQL statement)

Suppose now that you want to add a condition to the SQL statement in the previous exercise (see Figure 113: Advanced: the completed statement (example 2) on page 408) to show only rows of the table PARTS with a part number greater than 12456:

1. Press the Exit function key (F3) to return to the syntax for the subselect clause.

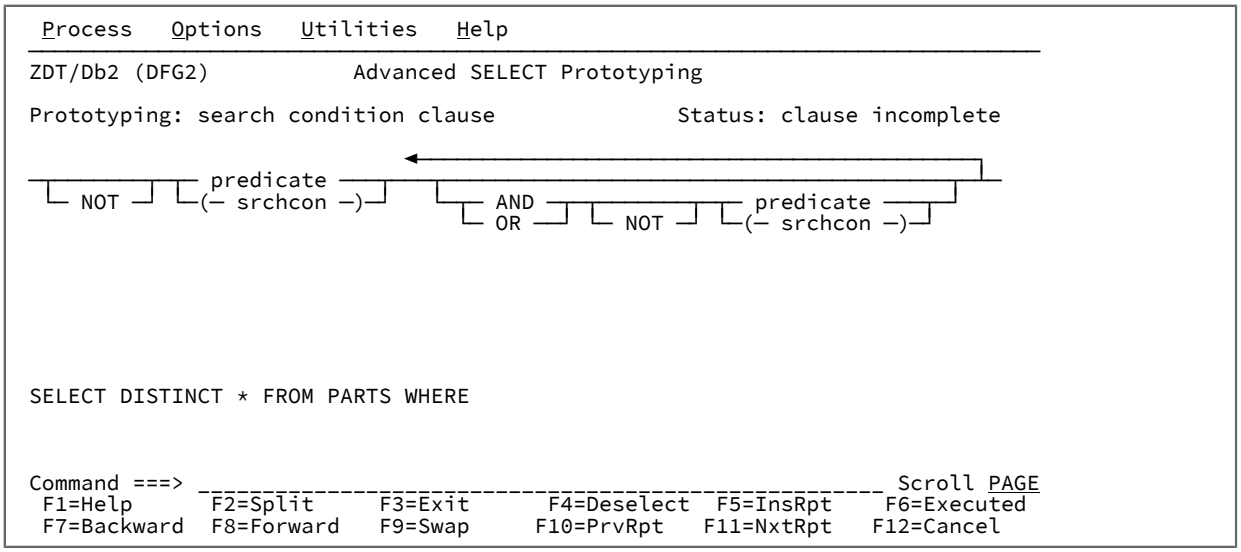


2. Move the cursor to the **where** fragment.
3. Press Enter. The syntax for the WHERE clause is displayed.

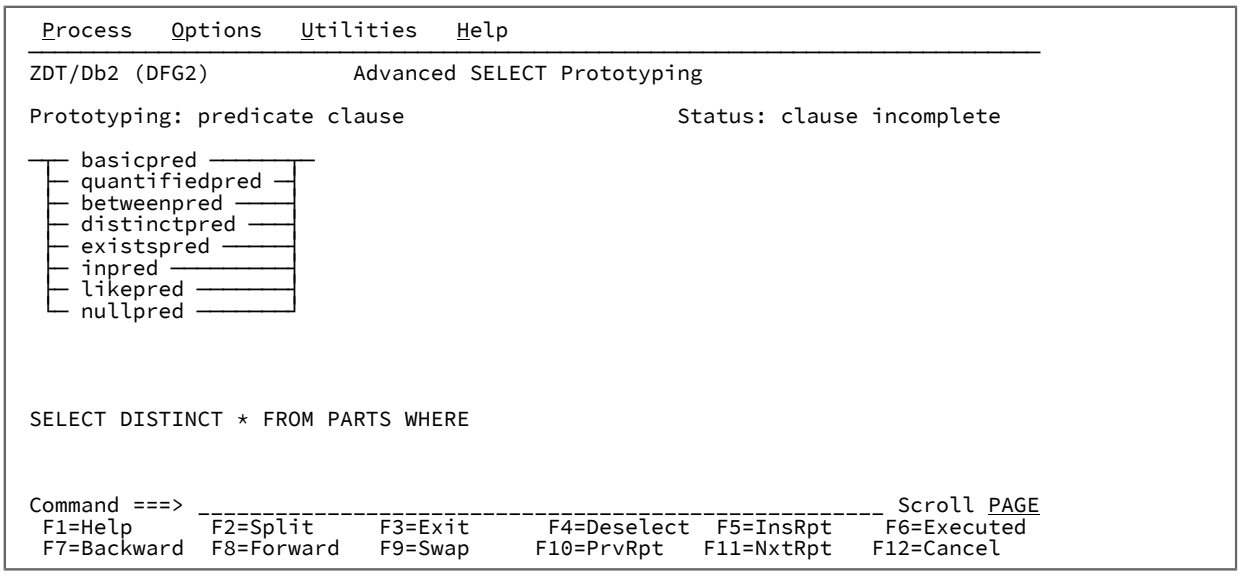


   Notice that ZDT/Db2 has added the required keyword WHERE to the SQL statement.

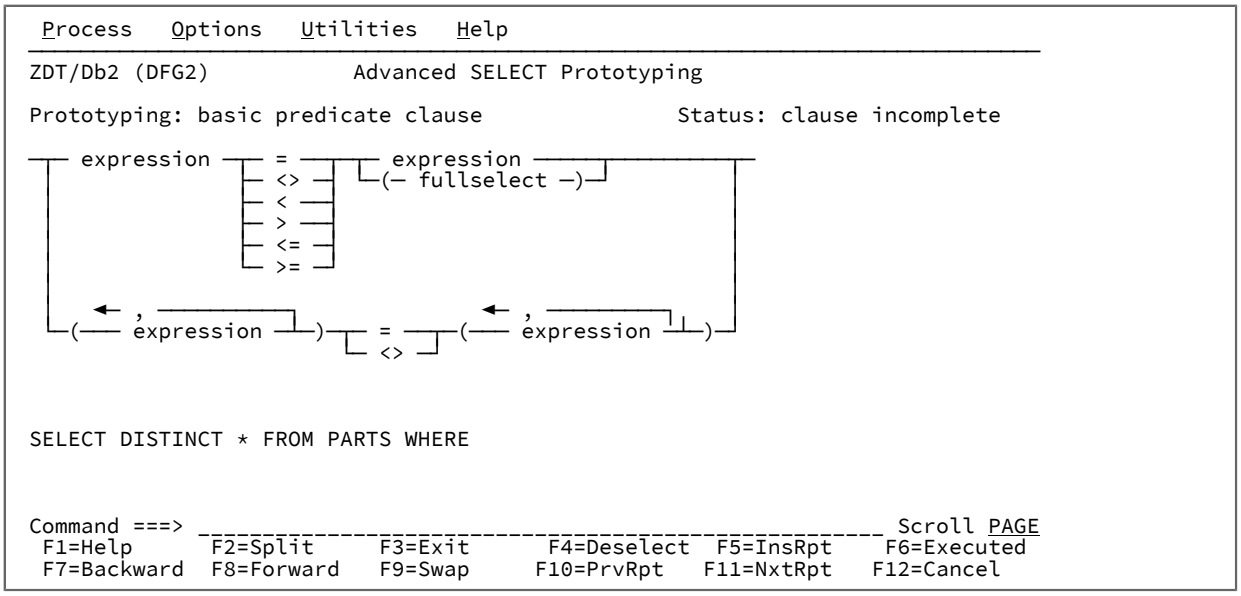
4. Move the cursor to the **srchcon** fragment.
5. Press Enter. The syntax for the search condition clause is displayed.



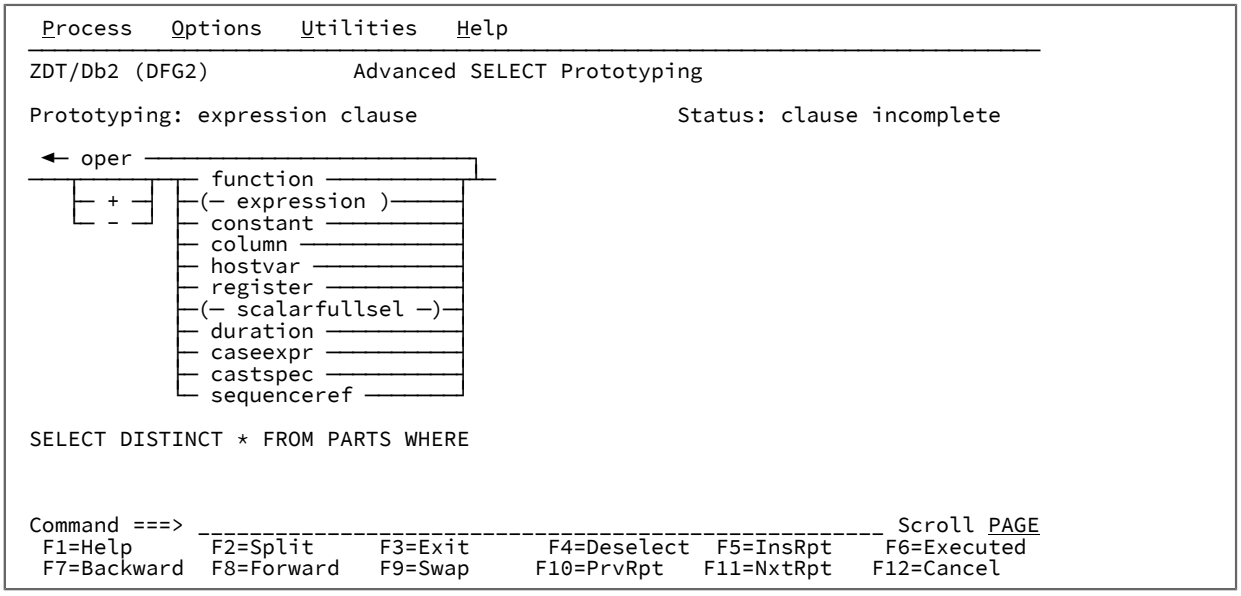
6. Move the cursor to the first **predicate** fragment.
7. Press Enter. The syntax for the predicate clause is displayed.



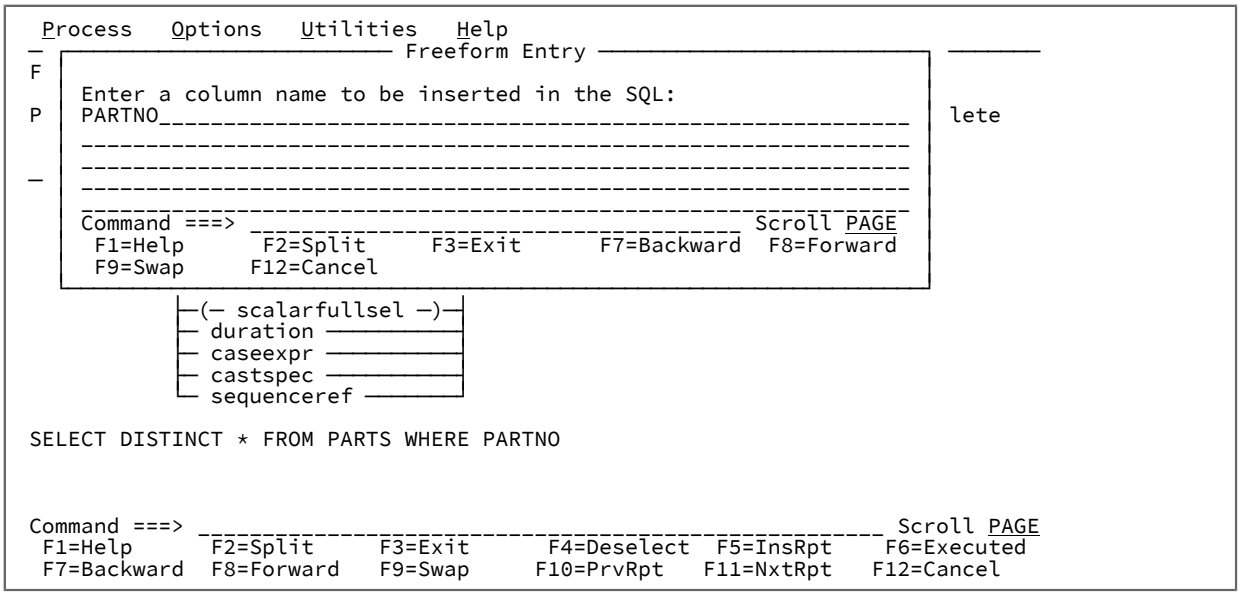
8. Move the cursor to the **basicpred** fragment.
9. Press Enter. The syntax for the basic predicate clause is displayed.



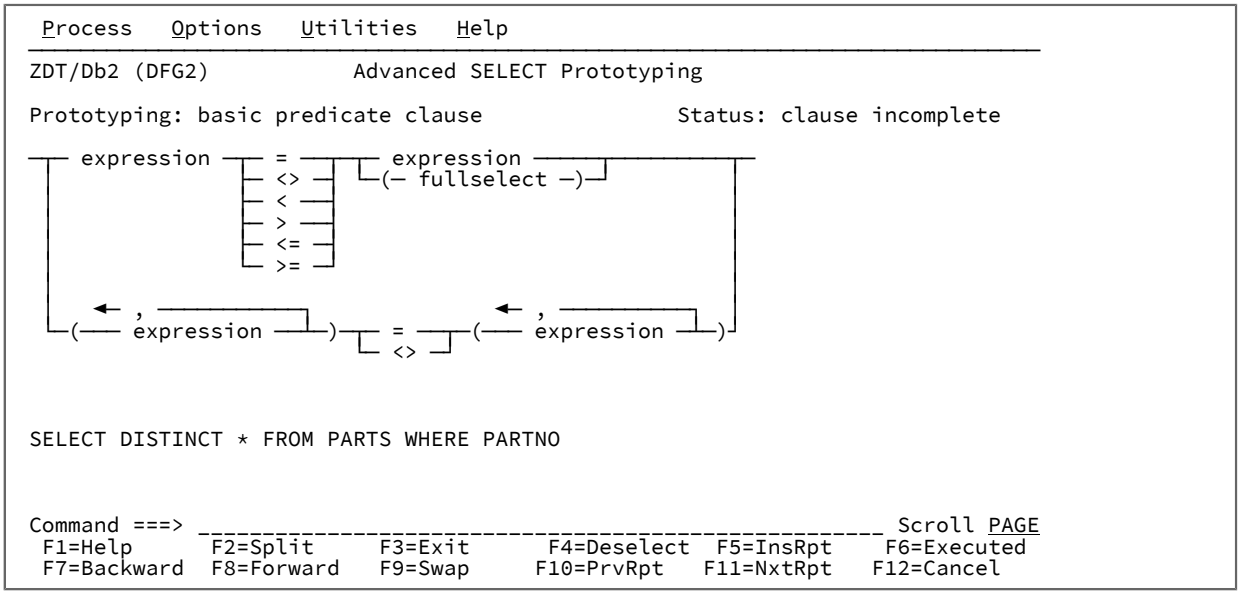
10. Move the cursor to the **expression** fragment (to the left of the list of operators).
11. Press Enter. The syntax for the expression clause is displayed.



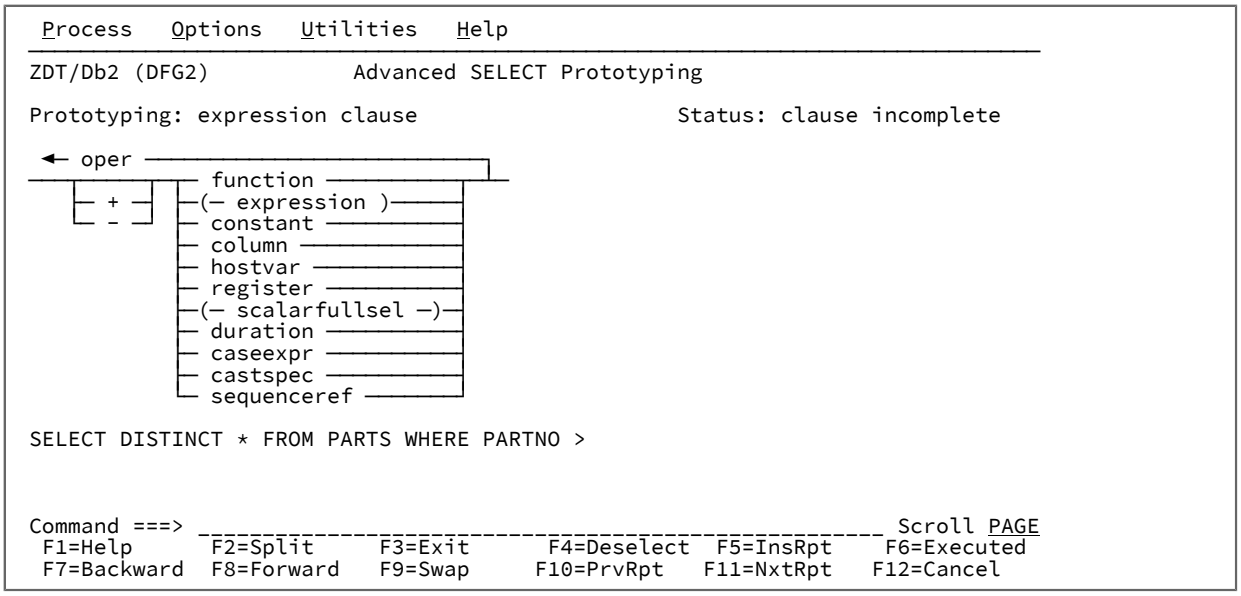
12. Move the cursor to the **column** fragment.
13. Press Enter. A pop-up panel is displayed in which you can enter a column name.
14. Type `PARTNO` in the pop-up panel.



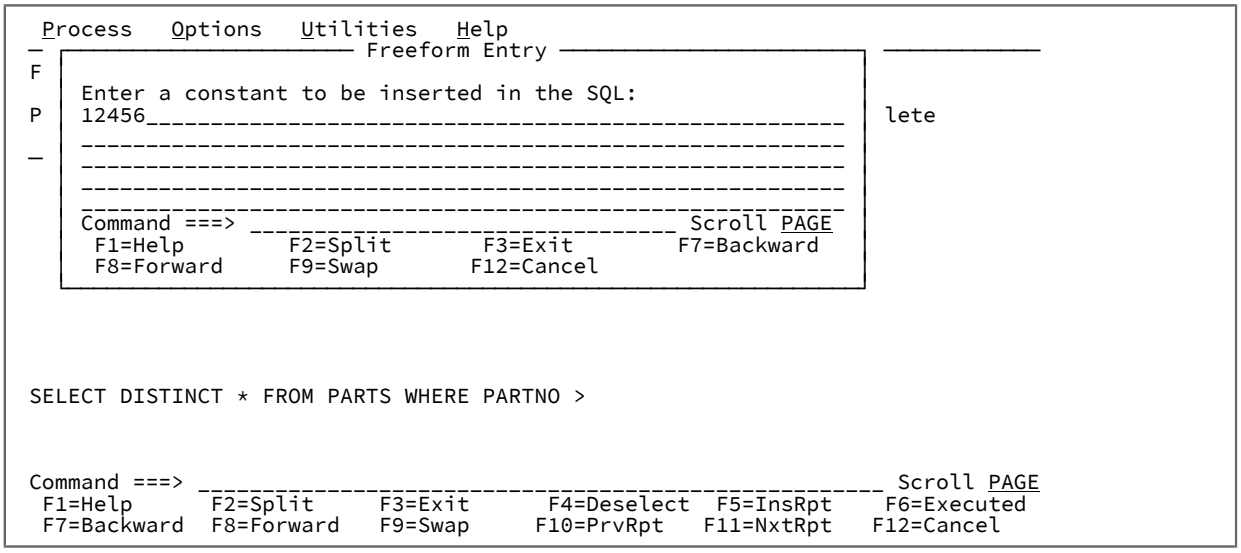
15. Press the Exit function key (F3). The column name PARTNO is added to the SELECT statement.
16. Press the Exit function key (F3). The syntax for the basic predicate clause is displayed.



17. Move the cursor to the *"greater than"* (>) symbol.
18. Press Enter. A *"greater than"* (>) symbol is displayed in the partly-built SELECT statement.
19. Move the cursor to the **expression** fragment (to the right of the list of operators).
20. Press Enter. The syntax for the expression clause is displayed.

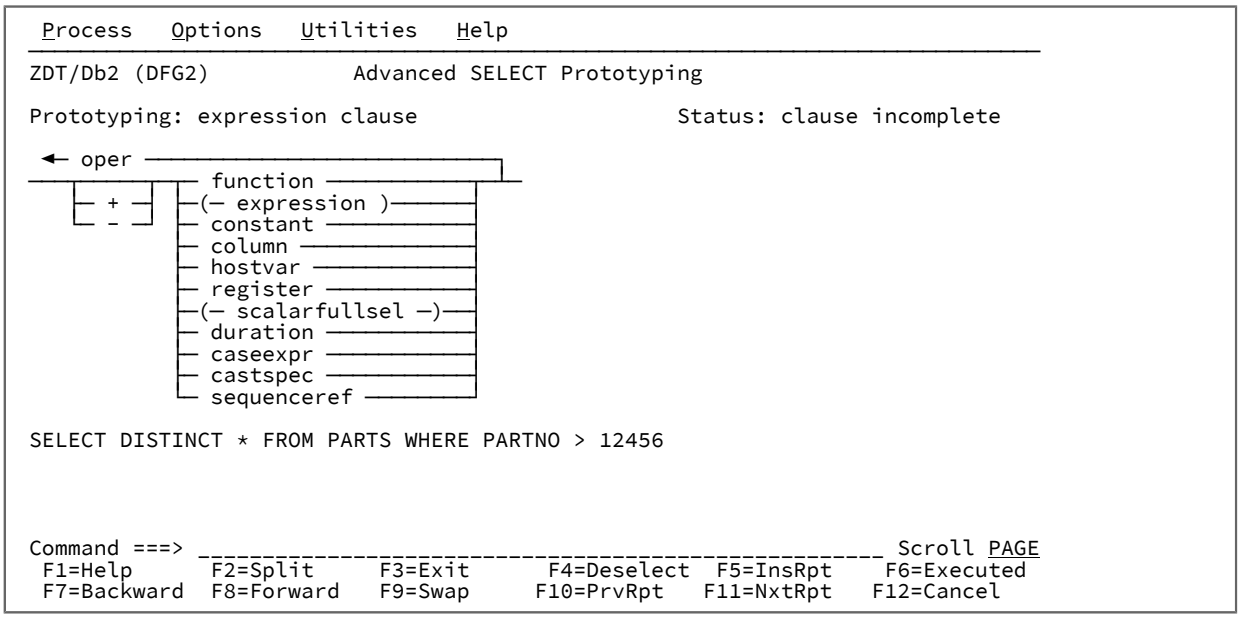


21. Move the cursor to the **constant** fragment.
22. Press Enter. A pop-up panel is displayed in which you can enter a constant.
23. Type `12456` in the pop-up panel.



24. Press the Exit function key (F3).

Figure 114. Advanced: the completed statement (example 3)



The panel shows the completed SELECT statement:

```
SELECT DISTINCT * FROM PARTS WHERE PARTNO > 12456
```

Related tasks

Example 2 (Changing the SQL statement) on page 407
Executing a SELECT statement after you have built it using prototyping on page 424

Example 4 (Removing part of the SQL statement)

To remove part of a SELECT statement you have built using advanced SELECT prototyping, use the Deselect function key (F4). The effect of the Deselect function key depends on which fragment of the syntax is currently active. For more information, see the description of the DESELECT command.

Suppose that you want to remove the condition you added to the SQL statement in the previous exercise (see Figure 114: Advanced: the completed statement (example 3) on page 414) that restricted the rows shown to those with a part number greater than 12456. This means that you need to remove the WHERE clause from the SELECT statement.

You can remove the WHERE clause by either of these methods:

- Using the Deselect function key (F4):
  1. Place the cursor on the WHERE clause in the generated SELECT statement.
  2. Press Enter to select the WHERE clause. It changes to green, and the syntax for the WHERE clause is redisplayed.
  3. Place the cursor on the command line.
  4. Press the Deselect function key (F4).
- Returning to the prototyping panel that contains the **where** fragment and *"deselecting"* it:

Note: The following steps assume the current panel on your screen is the last panel in Example 3 as shown in Figure 114: Advanced: the completed statement (example 3) on page 414.

1. Press the Exit function key (F3). ZDT/Db2 returns you to the syntax for the basic predicate clause.
2. Press the Exit function key (F3). ZDT/Db2 returns you to the syntax for the predicate clause.
3. Press the Exit function key (F3). ZDT/Db2 returns you to the syntax for the search condition clause.
4. Press the Exit function key (F3). ZDT/Db2 returns you to the syntax for the WHERE clause.
5. Press the Exit function key (F3). ZDT/Db2 returns you to the syntax for the subselect clause. This panel contains the **where** fragment.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           Advanced SELECT Prototyping

Prototyping: subselect clause                     Status: statement complete

— select — from ┌─────────┐ ┌─────────┐ ┌─────────┐
                └ where ─┘ └ groupby ─┘ └ having ─┘









SELECT DISTINCT * FROM PARTS WHERE PARTNO > 23456


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Deselect  F5=InsRpt    F6=Executed
 F7=Backward  F8=Forward   F9=Swap     F10=PrvRpt   F11=NxtRpt   F12=Cancel
```

6. Move the cursor to the **where** fragment.
7. Press the Deselect function key (F4).

Figure 115. Advanced: the statement with WHERE clause removed (example 4)

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           Advanced SELECT Prototyping

Prototyping: subselect clause                     Status: statement complete

— select — from ┌─────────┐ ┌─────────┐ ┌─────────┐
                └ where ─┘ └ groupby ─┘ └ having ─┘









SELECT DISTINCT * FROM PARTS


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Deselect  F5=InsRpt    F6=Executed
 F7=Backward  F8=Forward   F9=Swap     F10=PrvRpt   F11=NxtRpt   F12=Cancel
```

The panel shows the SELECT statement without the WHERE clause:

```
SELECT DISTINCT * FROM PARTS
```

Related tasks

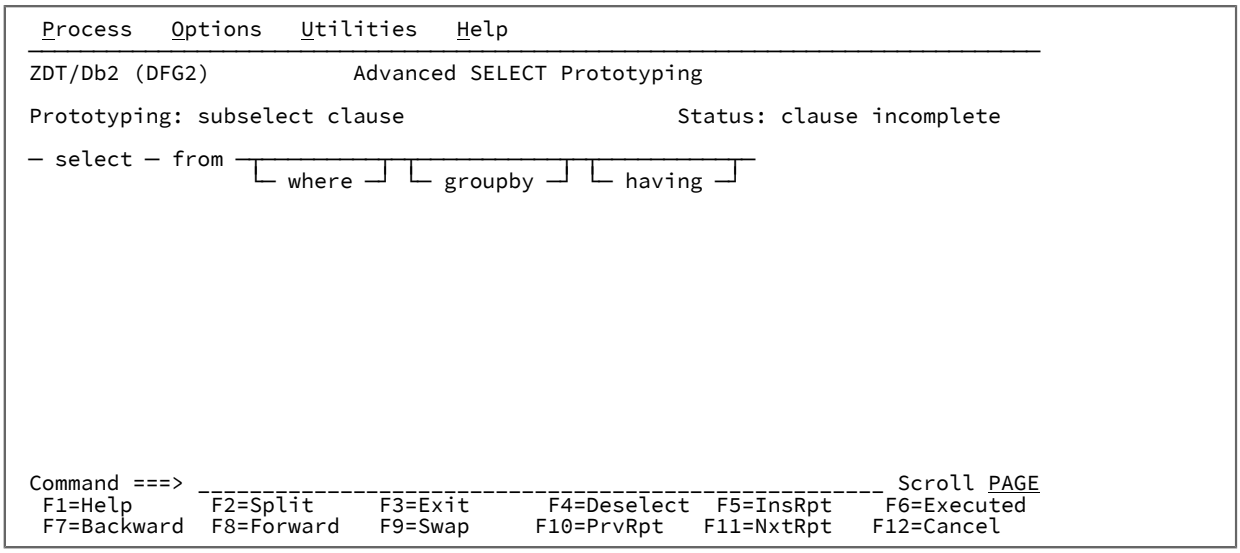
Example 3 (Adding to the SQL statement) on page 408
Executing a SELECT statement after you have built it using prototyping on page 424

Handling repeatable items in the syntax

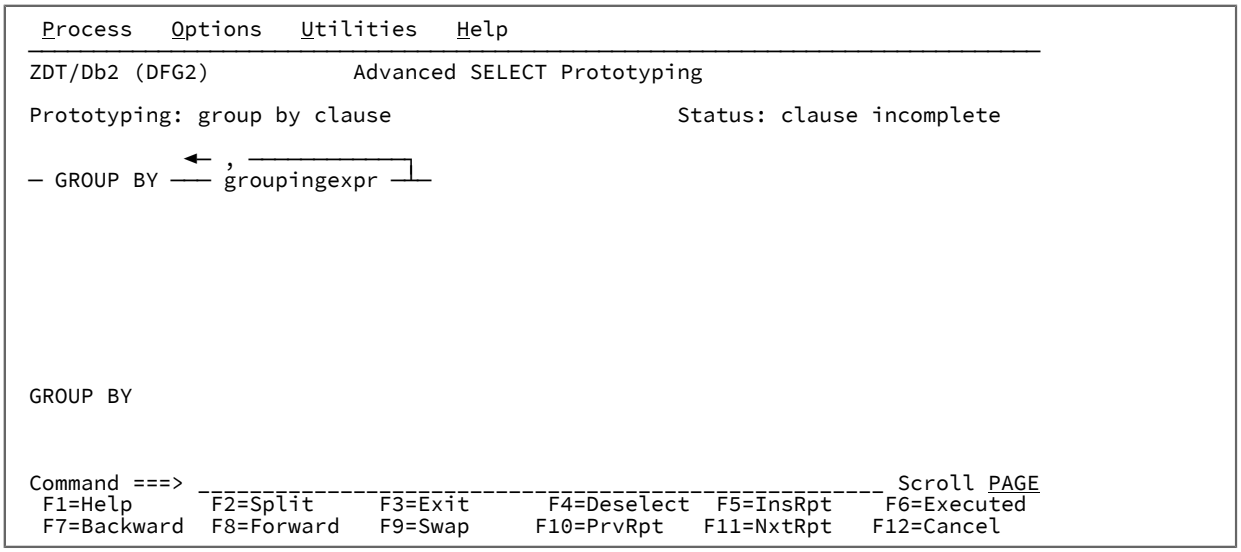
Some parts of the SELECT statement syntax contain repeatable items where one or more repeatable items are delimited by a repeat separator. The repeat separator for some repeat fragments is a comma (,), for some others it is a constant or an operand, and for others it is a space. If you specify more than one repeatable item, and the separator is constant, Advanced SELECT prototyping inserts the correct separator for you; otherwise, you must select the separator like any other element.

Let's take a simple example to show how you can build a fragment of the SELECT clause with repeatable items. Say you want to show certain information (specified in the rest of the SELECT statement) about a company's employees, grouped by their skill type, grade, and the year they started.

1. Repeat steps 1 to 5 of the first example (see Example 1 (Building a simple SQL statement) on page 402) so that the syntax for the subselect clause is displayed.



2. Move the cursor to the **groupby** fragment.
3. Press Enter. The syntax for the GROUP BY clause is displayed.

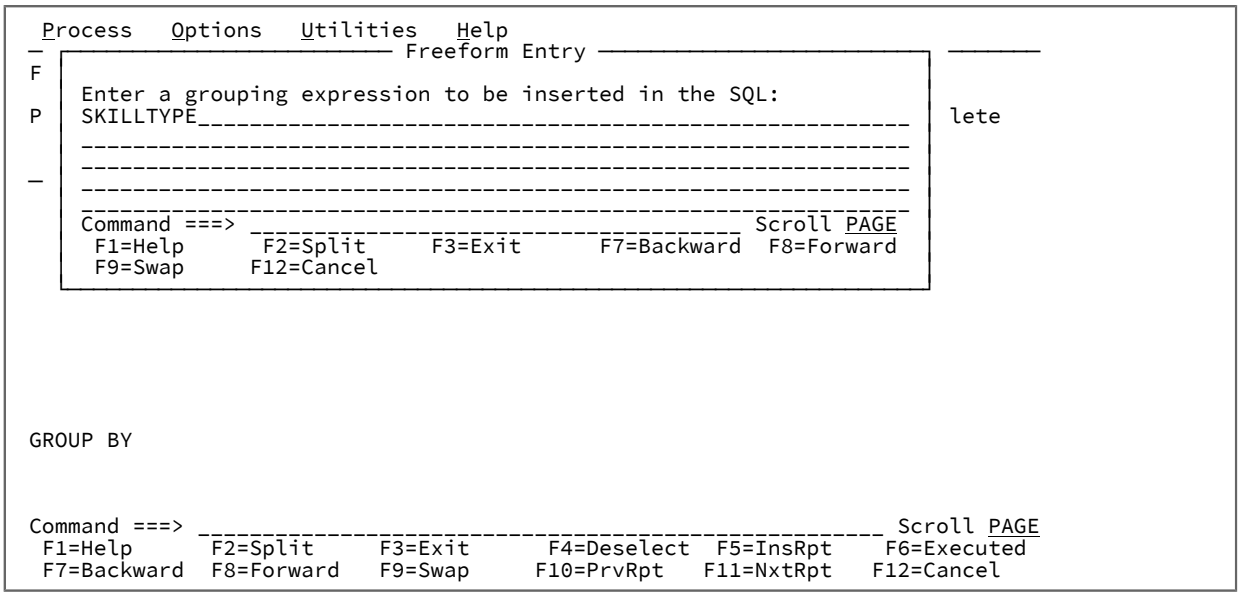


Notice that ZDT/Db2 shows the required keywords GROUP BY in the partly-built SQL clause on the panel.

You can see that the syntax for the GROUP BY clause consists of:

- The required key words GROUP BY (shown on your screen in white),
- The repeatable item **groupingexpr** (shown on your screen in red), and
- A comma (,) as the repeat separator.

4. Move the cursor to the **groupingexpr** fragment.
5. Press the InsRpt function key (F5). A pop-up panel is displayed in which you can enter a column name.
6. Type `SKILLTYPE` in the pop-up panel.



7. Press the Exit function key (F3). The column name SKILLTYPE is added to the GROUP BY clause.

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)           Advanced SELECT Prototyping

Prototyping: group by clause                      Status: clause complete

                ←  , ──────────┐
— GROUP BY ─── groupingexpr ──┴──




GROUP BY SKILLTYPE


Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Deselect  F5=InsRpt    F6=Executed
 F7=Backward  F8=Forward   F9=Swap     F10=PrvRpt   F11=NxtRpt   F12=Cancel
```

Notice that, at this stage, ZDT/Db2 has not added a repeat separator to the statement. If you are only specifying one repeatable item, this is the correct syntax. However, if ZDT/Db2 detects that you are specifying more than one repeatable item (as in the following steps), it inserts the repeat separator.

8. Again, move the cursor to the **groupingexpr** fragment.
9. Press the InsRpt function key (F5) to display the pop-up panel.
10. Type `EMPGRADE` in the pop-up panel.
11. Press the Exit function key (F3). The repeat separator and column name EMPGRADE are added to the GROUP BY clause.

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)           Advanced SELECT Prototyping

Prototyping: group by clause                      Status: clause complete

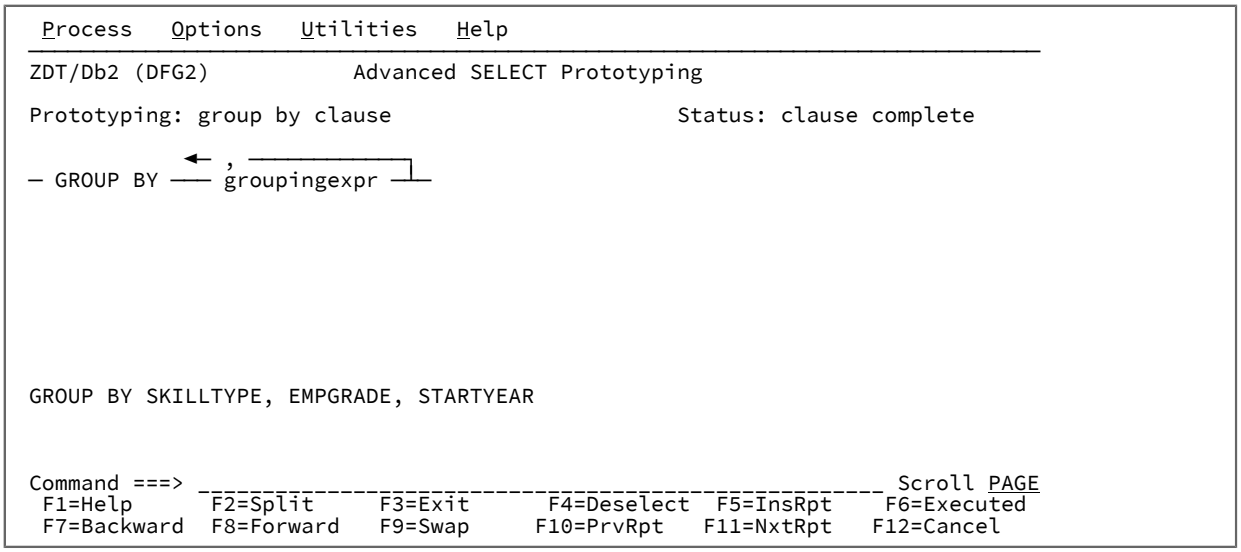
                ←  , ──────────┐
— GROUP BY ─── groupingexpr ──┴──




GROUP BY SKILLTYPE , EMPGRADE


Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Deselect  F5=InsRpt    F6=Executed
 F7=Backward  F8=Forward   F9=Swap     F10=PrvRpt   F11=NxtRpt   F12=Cancel
```

12. Repeat steps 8 on page 418 through 11 on page 418 for STARTYEAR.



The panel shows the completed GROUP BY clause:

```
GROUP BY SKILLTYPE , EMPGRADE , STARTYEAR
```

Note: If you only require one repeatable item in the fragment, after you move the cursor to the **groupingexpr** fragment (step 4 on page 417), you can press the InsRpt function key (F5) or Enter.

Related tasks

Example 1 (Building a simple SQL statement) on page 402
Navigating through the repeatable items on page 419
Inserting an extra repeatable item on page 420
Changing a repeatable item on page 420
Deleting a repeatable item on page 421
Handling selectable repeat separators on page 422
Points to consider when working with repeatable items on page 423

Navigating through the repeatable items

After you have added a number of repeatable items to a clause, you might need to modify part of the clause by inserting an extra repeatable item, or changing or deleting an existing repeatable item. In any of these cases, you need to navigate your way through the repeatable items so that the appropriate repeatable item is current.

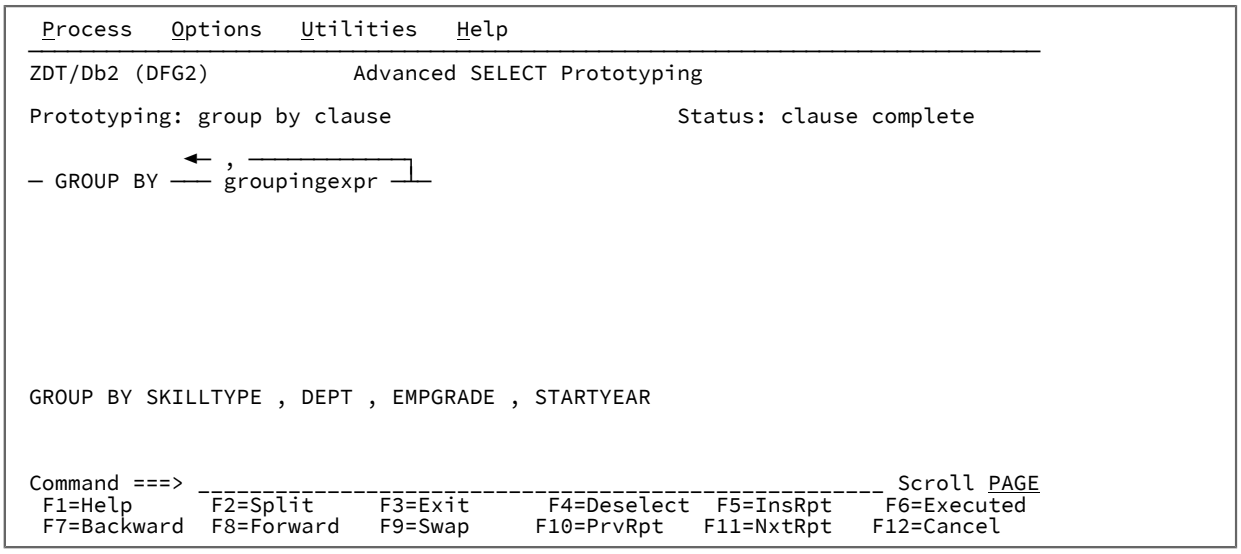
To make a repeatable item current (and therefore change its color from yellow to green), either:

- Move the cursor to one of the elements within the repeat group in the syntax, and use PrvRpt (F10) to scroll backwards or NxtRpt (F11) to scroll forwards through the repeatable items in the generated statement, or
- Move the cursor to the repeatable item in the generated statement, and press Enter.

Inserting an extra repeatable item

Suppose you need to insert an extra repeatable item (department) in the previous GROUP BY clause so that the information is grouped by their skill type, *department*, level, and the year they started:

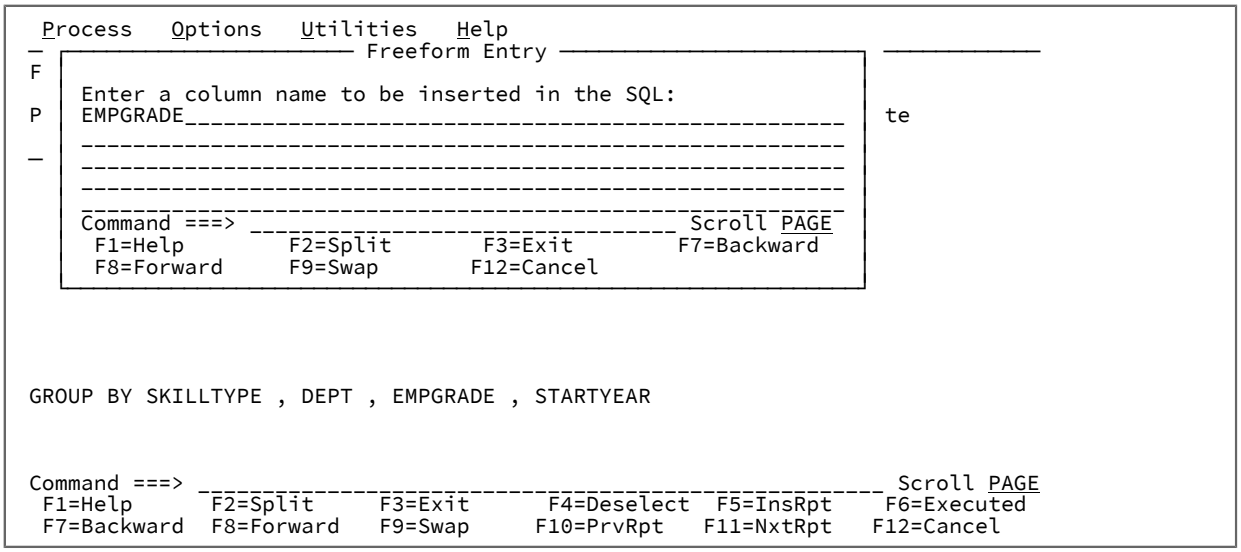
1. Move the cursor to the **SKILLTYPE** repeatable item (that is to precede the new repeatable item) in the GROUP BY clause.
2. Press Enter to make **SKILLTYPE** the current repeatable item. Its color changes from yellow to green.
3. Move the cursor to the **groupingexpr** fragment.
4. Press the InsRpt function key (F5). A pop-up panel is displayed in which you can enter a column name.
5. Type `DEPT` in the pop-up panel.
6. Press the Exit function key (F3). The repeat separator and the new repeatable item, DEPT, are added to the GROUP BY statement after the SKILLTYPE repeatable item.



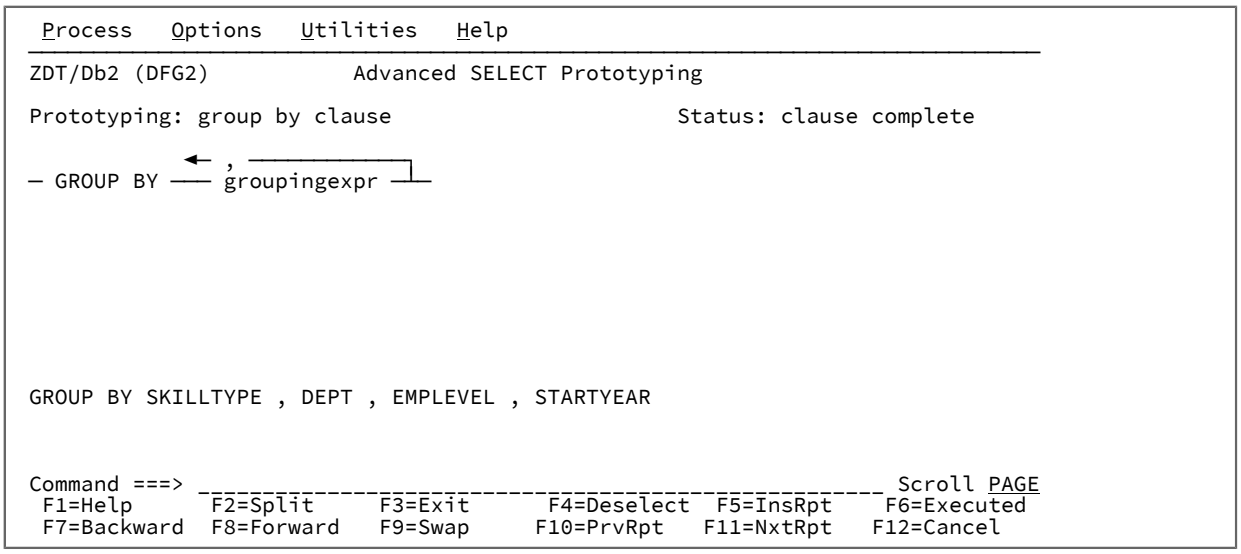
Changing a repeatable item

Suppose you want to change the name of the repeatable item, EMPGRADE, in the previous GROUP BY clause to EMPLEVEL:

1. Move the cursor to the **groupingexpr** fragment.
2. Press the NxtRpt function key (F11) to make EMPGRADE the current repeatable item. Its color changes from yellow to green.
3. Press Enter. A pop-up panel is displayed showing the repeatable item EMPGRADE already entered.



4. You can now edit the name of the column in the pop-up panel. Change it to `EMPLEVEL`.
5. Press the Exit function key (F3). EMPGRADE is replaced by EMPLEVEL in the GROUP BY statement.



Deleting a repeatable item

Suppose you want to remove the repeatable item, STARTYEAR, from the previous GROUP BY clause:

1. Move the cursor to the repeatable item, STARTYEAR
2. Press Enter to make STARTYEAR the current repeatable item. Its color changes from yellow to green.
3. Move the cursor to the **groupingexpr** fragment.

4. Press the Deselect function key (F4). The repeatable item, STARTYEAR (and its preceding repeat separator), is deleted from the GROUP BY clause.

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)           Advanced SELECT Prototyping

Prototyping: group by clause                      Status: clause complete

                ◄─ , ──────────┐
─ GROUP BY ──── groupingexpr ──┴─




GROUP BY SKILLTYPE , DEPT , EMPLEVEL


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Deselect  F5=InsRpt    F6=Executed
 F7=Backward  F8=Forward   F9=Swap     F10=PrvRpt   F11=NxtRpt   F12=Cancel
```

Handling selectable repeat separators

Some repeatable items in the SELECT statement syntax have a repeat separator that you can select (rather than a constant such as a comma). For example, the expression clause as shown in Figure 116: Example of a selectable repeat separator on page 422 uses an operator (**oper**) as a repeat separator.

Figure 116. Example of a selectable repeat separator

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)           Advanced SELECT Prototyping

Prototyping: expression clause                    Status: clause incomplete

 ◄─ oper ──────────────────────┐
──┬──────┬──┬─ function ──────┬┴─
  ├─ + ─┤  ├─(─ expression ─)─┤
  └─ - ─┘  ├─ constant ────────┤
           ├─ column ──────────┤
           ├─ hostvar ─────────┤
           ├─ register ────────┤
           ├─(─ scalarfullsel ─)─┤
           ├─ duration ────────┤
           ├─ caseexpr ────────┤
           ├─ castspec ────────┤
           └─ sequenceref ─────┘

SELECT ALL



Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Deselect  F5=InsRpt    F6=Executed
 F7=Backward  F8=Forward   F9=Swap     F10=PrvRpt   F11=NxtRpt   F12=Cancel
```

To show how to handle selectable repeat separators, we'll use an example. To generate the following expression clause:

```
AMT1 + AMT2 - AMT3
```

where the columns `AMT1`, `AMT2`, and `AMT3` are three repeat items, and "`+`" and "`-`" are the two repeat separators, perform the following steps:

1. Move the cursor to the **column** element and press Enter. A pop-up panel is displayed.
2. Type `AMT1` and press the InsRpt function key (F5). AMT1 is added to the clause:

   ```
   AMT1
   ```

3. Move the cursor to the **oper** repeat separator and press the InsRpt function key (F5). A list of valid repeat separators is displayed.
4. Move the cursor to the *"+"* repeat separator and press Enter. *"+"* is added to the clause:

   ```
   AMT1 +
   ```

5. Press the Exit function key (F3). The syntax for the expression clause is redisplayed.
6. Move the cursor to the **column** element and press Enter. A pop-up panel is displayed.
7. Type `AMT2` and press the InsRpt function key (F5). AMT2 is added to the clause:

   ```
   AMT1 + AMT2
   ```

8. Move the cursor to the **oper** repeat separator and press the InsRpt function key (F5). A list of valid repeat separators is displayed.
9. Move the cursor to the *"-"* repeat separator and press Enter. *"-"* is added to the clause:

   ```
   AMT1 + AMT2 –
   ```

10. Press the Exit function key (F3). The syntax for the expression clause is redisplayed.
11. Move the cursor to the **column** element and press Enter. A pop-up panel is displayed.
12. Type `AMT3` and press the InsRpt function key (F5). AMT3 is added to the clause:

    ```
    AMT1 + AMT2 – AMT3
    ```

Note: The repeat separator is associated with the repeat item it follows. When the last repeat item is current, you cannot edit the repeat separator.

Points to consider when working with repeatable items

If you are working with repeatable items, keep in mind the following:

- Repeatable item commands are only effective if you move the cursor to one of the elements of the repeatable item.
- For each repeatable item in a fragment (generally none or one), only one instance of the item is current at a given time. Generated data for the current instance is displayed in green; generated data for any other instances is displayed in yellow.
- Press the PrvRpt (F10) and NxtRpt (F11) function keys to navigate between the instances.
- The InsRpt function key (F5) inserts a repeat instance after the active repeat instance and makes the new repeat instance active.
- If there are no instances for a repeatable item, you can create the initial instance and make it active by either:
  - Using the InsRpt function key (F5), or
  - Moving the cursor to an element of the repeatable item and pressing Enter
- If there is already an instance of a repeatable item, pressing Enter with the cursor on an element of the repeatable item causes that element to be "edited?.

Executing a SELECT statement after you have built it using prototyping

When you have completed building an SQL SELECT statement, using either basic SELECT prototyping or advanced SELECT prototyping, you can run it by:

- Entering the EXECUTE primary command, or
- Pressing the Execute function key (F6)

ZDT/Db2 displays the results according to the **Use edit (instead of browse) to display results** option. If this option is selected, ZDT/Db2 displays the results using ZDT/Db2 Edit; otherwise, ZDT/Db2 displays the results using ZDT/Db2 Browse.

Figure 117: Sample results table displayed using Browse on page 424 shows a results table using ZDT/Db2 Browse.

Figure 117. Sample results table displayed using Browse

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)               Select Statement Browse
RESULTS FOR SELECT #1.FIRSTNME, #1.LASTNAME, #1.JOB FROM D...      Format TABL
FIRSTNME     LASTNAME        JOB
#2           #4              #8
VARCHAR(12)  VARCHAR(15)     CH(8)
<---+----1-> <---+----1----> <---+-->
****  Top of data  ****
MICHAEL<     THOMPSON<       MANAGER
IRVING<      STERN<          MANAGER
EVA<         PULASKI<        MANAGER
THEODORE<    SPENSER<        MANAGER
VINCENZO<    LUCCHESI<       SALESREP
SEAN<        O'CONNELL<      CLERK
DOLORES<     QUINTANA<       ANALYST
HEATHER<     NICHOLLS<       ANALYST
ELIZABETH<   PIANKA<         DESIGNER
MASATOSHI<   YOSHIMURA<      DESIGNER
MARILYN<     SCOUTTEN<       DESIGNER
JAMES<       WALKER<         DESIGNER
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

If there is insufficient space on the panel to show all of the columns, use the LEFT primary command to scroll left, or the RIGHT primary commands to scroll right, to view the additional columns.

Related tasks

ZDT/Db2 system options (option 0.2) on page 65

Related references

EXECUTE primary command on page 948

Entering, executing, and explaining SQL statements

If you want to:

- Enter freeform SQL statements,
- Run freeform SQL statements, or
- Obtain information about an SQL statement

select option 3 (Enter) on the SQL Prototyping, Execution and Analysis panel.

When you select this option, ZDT/Db2 displays the Enter, Execute and Explain SQL Statements panel.

Related tasks

Entering SQL statements on page 425
Executing SQL statements on page 426
Explaining SQL on page 427

Related references

SQL Prototyping, Execution and Analysis panel on page 843
Enter, Execute and Explain SQL Statements panel on page 684

Entering SQL statements

You can enter SQL statements freeform in the SQL statement entry area. If you want to enter more than one SQL statement, you must separate the statements with a semicolon (;). When the SQL statements are processed, the last column of each line is joined directly to the first column of the next line. If a token ends in the last column of a line, ensure that the first column of the next line contains a blank or a delimiter token.

For detailed information about constructing SQL statements, see the *DB2 for z/OS SQL Reference*.

Editing SQL

You can edit the SQL you have entered by entering the SQL primary command. The SQL you have entered is copied to a temporary data set and an ISPF edit session is started. When you complete the Edit session, the temporary data set is copied back to the entry area on the Enter, Execute and Explain SQL Statements panel.

When you edit the SQL, you can:

- Save it to a data set for later use (using the ISPF edit CREATE command).
- Retrieve SQL you have saved previously (using the ISPF edit COPY command).
- Exploit the power of the ISPF editor to construct your SQL.

Related references

Enter, Execute and Explain SQL Statements panel on page 684
SQL primary command on page 1005

Limiting the rows loaded into the ZDT/Db2 editor

To limit the rows loaded into the ZDT/Db2 editor when the SQL statement being processed is a SELECT statement, specify the maximum number of rows in the **Row count** field on the Enter, Execute and Explain SQL Statements panel.

To load all rows for the SELECT statement, specify * or ALL in this field.

Scrolling the entry area

If the statement you are entering extends beyond the visible display area, you can scroll through your data, using the following function keys:

Up function key (F7)

Scroll backward (up)

Down function key (F8)

Scroll forward (down)

You can also scroll using the primary commands UP and DOWN.

To control how far you scroll when you press one of the scrolling function keys, enter a scroll amount in the **Scroll** field:

Scroll amount

Scrolls…

PAGE

One page of data at a time

HALF

Half a page of data at a time

MAX

To the top or bottom

DATA

One line less than a page of data at a time

CSR

To the cursor position (if the cursor is within the data, otherwise the default scroll amount, PAGE, applies)

nnnn

nnnn lines at a time.

Related references

Enter, Execute and Explain SQL Statements panel on page 684

Executing SQL statements

To run the SQL you have entered in the statement entry area, press Enter.

If the SQL statement returns a result table, ZDT/Db2 displays it in an edit or browse panel, depending on the setting of the **Use edit (instead of browse) to display results** option.

If you have entered more than one SQL statement, the statements are run in sequence, interrupted by the display of any result tables. Execution continues until:

- An error occurs,
- The CANCEL command is entered on a result table display, or
- All the statements have been run

You can use the Enter, Execute and Explain SQL Statements panel to specify one or more of the following SQL SELECT statements (without using prototyping) and then run them in sequence:

```
ALTER                COMMENT ON           COMMIT
CREATE               DELETE               DROP
EXPLAIN              GRANT                INSERT
LABEL ON             LOCK TABLE           RENAME
REVOKE               ROLLBACK             SELECT
SET CURRENT DEGREE   SET CURRENT SQLID    UPDATE
```

Related tasks

Entering SQL statements on page 425
ZDT/Db2 system options (option 0.2) on page 65

Related references

Enter, Execute and Explain SQL Statements panel on page 684

SQL cost

If any SELECT statements are run, and no errors occur, ZDT/Db2 displays a message after each execution showing the total SQL *"cost"* of all the SELECT statements.

SQL errors

If an SQL error occurs while a statement is being run or explained, a panel is displayed containing the SQL error information.

To display the full text of the SQL statement that caused the error, press the SQL function key (F5) or enter `SQL` on the command line.

When you return from the error panel, ZDT/Db2 places the cursor on the failing statement or, if a syntax error occurred, places the cursor on the token in the failing statement that Db2® identified as in error. Statements that ran successfully before the failing statement are not rolled back, and no further statements are run or explained.

Explaining SQL

To explain the SQL you have entered in the statement entry area, enter the EXPLAIN primary command. ZDT/Db2 creates an EXPLAIN statement for the entered SQL, and optionally displays the Plan Table Rows panel showing the rows added by Db2® as a result of the explain.

Note: For detailed information about explaining an SQL statement, see the *DB2 for z/OS SQL Reference*.

To help identify the results of this explanation, you can enter a number in the **Query number** entry field. If you leave **Query number** blank, ZDT/Db2 generates a query number for you in the format YYYYMMDD*ss*, where *ss* is a sequence number.

If you have selected the **Show results** option, ZDT/Db2 attempts to display the relevant rows that Db2® has inserted in the plan table as a result of explaining the statement. If you have entered more than one SQL statement, the statements are explained in sequence, interrupted by the display of the plan table rows. Execution continues until an error occurs, the CANCEL command is entered on a plan table display, or all the statements have been explained.

You can use the Enter, Execute and Explain SQL Statements panel to explain the following SQL statements:

SELECT
INSERT
UPDATE (searched form only)
DELETE (searched form only)

The following primary commands are available with the Enter, Execute and Explain SQL Statements panel:

- DOWN
- EXECUTE
- EXPLAIN
- SQL
- UP

Related tasks

Entering SQL statements on page 425
Executing SQL statements on page 426
Additional considerations for EXPLAIN on page 429

Related references

Enter, Execute and Explain SQL Statements panel on page 684
Plan Table Rows panel on page 754
DOWN primary command on page 939
EXECUTE primary command on page 948
EXPLAIN primary command on page 950
SQL primary command on page 1005
UP primary command on page 1009

Additional considerations for EXPLAIN

The EXPLAIN primary command constructs and runs an SQL EXPLAIN statement. For the explain statement to run successfully, a *plan table* (called PLAN_TABLE) must exist with an owner name matching your current SQL authorization ID. Db2® places the results of the EXPLAIN statement in the plan table.

ZDT/Db2 also places relevant data in a *statement table* (called DSN_STATEMNT_TABLE) and a *function table* (called DSN_FUNCTION_TABLE) if they exist.

You can access the statement and function table rows related to a plan table row by using line commands.

Related tasks

Explaining SQL on page 427
Managing tables used by SQL explain on page 430

Related references

COL primary command on page 934
PLAN primary command on page 986
HINT primary command on page 968

Editing and executing SQL statements from a data set

If you want to:

- Optionally initiate an ISPF Edit session on a data set or PDS member containing SQL statements
- Run the SQL statements in a data set or PDS member

select option 4 (Edit) on the SQL Prototyping, Execution and Analysis panel.

Note: You cannot use this option to create a new member.

When you select this option, ZDT/Db2 displays the Edit and Execute SQL Statements from a Data Set panel.

To use this panel:

1. Specify the name of the data set or PDS member containing the SQL statements you want to edit. Use a combination of the **Data set name**, **Member**, and **Volume serial** entry fields.
2. To limit the rows loaded into the ZDT/Db2 editor when the SQL statement being processed is a SELECT statement, specify the maximum number of rows in the **Row count** field on the Edit and Execute SQL Statements from a Data Set panel.

   To load all rows for the SELECT statement, specify * or ALL in this field.
3. Set the processing options.
4. Press Enter.

If you selected the **Edit data set** processing option, the contents of the specified data set or PDS member are displayed in an ISPF Edit session. When you have completed editing the SQL statements, press the Exit function key (F3) to run the commands (the **Execute SQL from data set** processing option must be selected).

If you selected the **Execute SQL from data set** processing option, but not the **Edit data set** processing option, ZDT/Db2 immediately runs the SQL commands in the specified data set or PDS member

Note: If an SQL error occurs, to prevent ZDT/Db2 executing further SQL statements, use the Cancel function key (F12) to return you to the Edit and Execute SQL Statements from a Data Set panel.

Related tasks

Specifying a data set and a member name on page 52

Related references

SQL Prototyping, Execution and Analysis panel on page 843
Edit/Execute SQL (Data Set) panel on page 641

Managing tables used by SQL explain

If you want to select functions to support the use of the SQL EXPLAIN statement, select option 5 (Utility) on the SQL Prototyping, Execution and Analysis panel.

When you select this option, ZDT/Db2 displays the Explain Utilities panel. This panel lets you create and delete the tables used by the SQL EXPLAIN statement. For additional information about the tables used by the SQL EXPLAIN statement, see the *DB2 for z/OS SQL Reference*.

Use the Explain Utilities panel to specify the type of function you want to perform. For more details about using this panel, refer to the reference sections.

Related tasks

Entering, executing, and explaining SQL statements on page 424

Related references

SQL Prototyping, Execution and Analysis panel on page 843
Explain Utilities panel on page 686

Exiting SQL prototyping

To exit SQL prototyping, issue the CANCEL primary command or press the Cancel function key (F12).

Related references

CANCEL primary command on page 926

Chapter 14. Using ZDT/Db2 functions in batch jobs

You can perform ZDT/Db2 functions that can be executed in batch jobs.

ZDT/Db2 functions that can be executed in batch jobs.

Edit models are provided that insert predefined statements and instructions for a selected Z Data Tools/Db2 function or REXX external function.

JCL syntax for Z Data Tools/Db2 batch jobs

To use a Z Data Tools/Db2 function in a batch job, include the following statement: `//stepname EXEC PGM=HFMDB2,PARM=('SSID=sysid,SQID=sqlid')"`

where,

sysid

Refers to the Db2 subsystem where the ZDT/Db2 functions run.

sqlid

Refers to the Db2 SQLID under which the ZDT/Db2 functions run.

By default, report output is written to SYSPRINT. Sections of the output can be directed instead to the following ddnames:

HFMRPRM

$$FILEM command parameters and return codes

HFMREPT

Command reports and return codes

If HFMPRINT is present in the JCL, it is used instead of SYSPRINT. If HFMIN is present in the JCL, it is used instead of SYSIN.

You might also need a STEPLIB statement or other DD statements, depending on which Z Data Tools functions the batch job contains.

When you code Z Data Tools control statements, you must:

- Begin each control statement with `$$FILEM` in columns 1 to 7, followed by a blank in column 8, followed by the command name, followed by the first parameter (if any).
- Specify parameters and their values separated by an equal sign, as follows:

  ```
  PARAMETER=value
  ```

- Follow each parameter (except the last one) immediately with a comma.

Comment statements, which begin with an asterisk (*), are also allowed.

To continue a control statement onto the next line, end the first line with a comma and begin the next line with `$$FILEM` and a blank.

To continue a parameter that does not fit onto one line, do not use a continuation character. Instead, split the parameter into more than one line, enclosing each part in apostrophes. For example:

```
$$FILEM OBJIOWNR='LONGOWNER12345678901234567890123456789012',
$$FILEM '2345678901234567890123456789012345678901NAME',
```

Notes:

1. If the control statement contains only the function name, there must be a space between the function name and the comma.
2. To change processing options in batch or to reset options to the installation default, use the SET function.

Example

Example

The following example shows a batch job that uses the ZDT/Db2 DSP function to print the Db2 table:

```
//jobcard
$$FILEM DBP FORMAT=TABLE,
$$FILEM OBJIOWNR='LONGOWNER12345678901234567890123456789012',
$$FILEM '2345678901234567890123456789012345678901NAME',
$$FILEM OBJINAME='DUMMY_TABLE_WITH_A_LO2345678901234567890',
$$FILEM '12345678901234567890123456789012345LONG_NAME',
$$FILEM UCREAD=YES,
$$FILEM NULLIND=_,
$$FILEM ROWS=ALL
```

Using Z Data Tools/Db2 edit models

This topic explains how to use edit models to create batch JCL to access Z Data Tools/Db2 functions.

The ISPF environment of your site must be configured to enable access to the Z Data Tools edit models. See *"Implementing Z Data Tools edit models in ISPF Edit"* in the *Z Data Tools Customization Guide*.

Edit models

An ISPF edit model is a predefined set of statements. You can include these statements in a member that you are editing and then modify them to suit your needs. Along with the edit models that are shipped with ISPF, Z Data Tools includes edit models for the Z Data Tools Base component, ZDT/Db2, and ZDT/IMS functions and REXX external functions.

An edit model has two parts:

Data lines

These are the actual lines that are placed in the data you are editing. In the Z Data Tools ISPF models the data lines describe the format of function parameters.

Notes

Notes provide tutorial information about the function parameters. These describe how to update the data lines to your requirements.

When you enter the MODEL command, you can select the Z Data Tools function for the data type (JCL or EXEC) of the member that is being edited.

Using edit models

If you edit a member of a data set of type JCL or EXEC and then issue the MODEL command, a list of applicable Z Data Tools functions or REXX external functions is displayed.

JCL

The **Z Data Tools Models** panel lists the available edit models, grouped by category.

You can enter RF to display another panel which lists the available edit models for REXX external functions.

EXEC

The **REXX Models** panel is displayed.

Enter FF to display the **Z Data Tools Rexx Function Models** panel, which lists the available edit models for REXX external functions.

Enter the function name to insert the statements from that edit model in the edit session. For example, enter DSEB to insert the Batch Edit model. Use the A (after) or B (before) line command to insert the statements at a specific location.

Issue the RESET command to permanently remove the note lines, leaving only the data lines.

Edit model example: JCL that calls the DBP function

This example shows how to use edit models to generate a batch job to print a Db2 table using the DBP function.

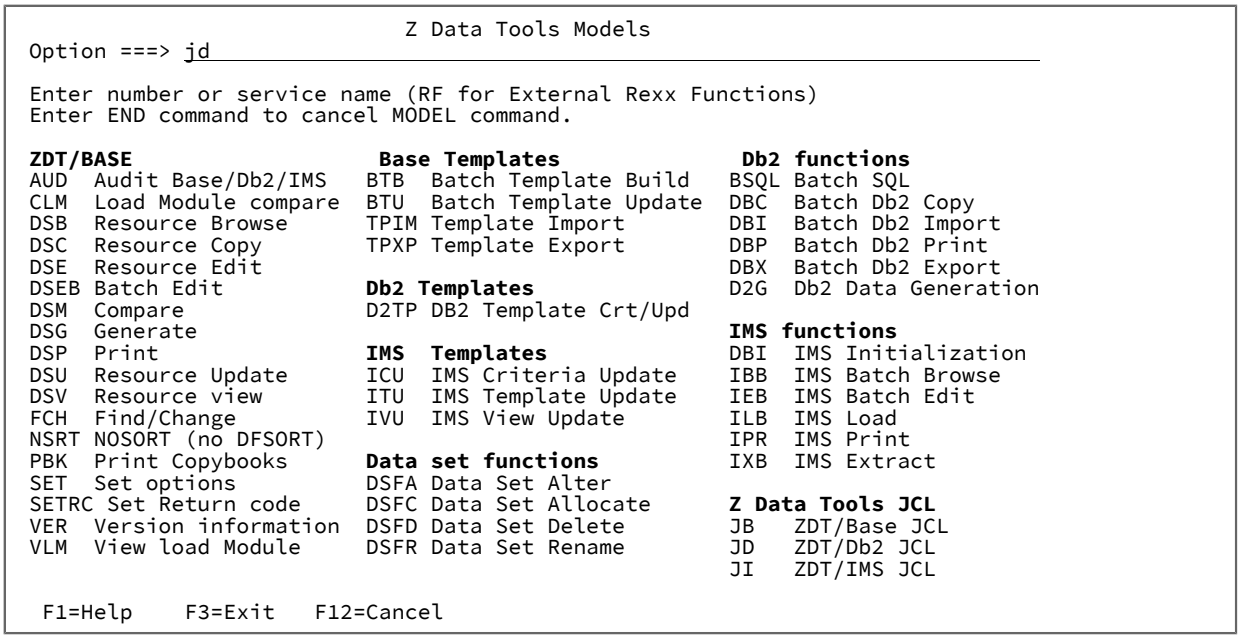
1. Select option 2 from the ISPF Primary Option Menu.
2. In the **Edit Entry Panel**, enter the name of a new member in a data set of type *"JCL"*. Example:

```
   Data Set Name . . . 'EXAMPLE.JCL(newdbpx)'
```

3. In the edit session, enter the MODEL primary command.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       EXAMPLE.JCL(NEWDBPX) - 01.00                Columns 00001 00072
Command ===> model                                      Scroll ===> CSR
****** ***************************** Top of Data ******************************
''''''
''''''
```

The **Z Data Tools Models** panel lists the available edit models in the JCL class.



4. Enter JD.

   The edit model for the ZDT/Db2 JCL is copied into the edit session.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       EXAMPLE.JCL(NEWDBPX) - 01.00                  Columns 00001 00072
Command ===> ____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ********************************
=NOTE=     ------------------------------------------------------
=NOTE=     --                                                  ---
=NOTE=     --  DB2 JCL                                         ---
=NOTE=     --                                                  ---
=NOTE=     ------------------------------------------------------
000001 //HFMDB2  EXEC PGM=HFMDB2,PARM=('SSID=sysid,SQID=sqlid')
000002 //SYSPRINT DD SYSOUT=*
000003 //HFMTSPRT DD SYSOUT=*
000004 //SYSOUT   DD SYSOUT=*
000005 //SYSTERM  DD SYSOUT=*
000006 //SYSTSPRT DD SYSOUT=*
=NOTE=
000007 //HFMODPP DD DISP=SHR,DSN=OPTIM.ODPP.SAMPLIB(ENVVARS)
=NOTE=
=NOTE=    HFMODPP DD
=NOTE=    To enable Z Data Tools to specify and run ODPP commands,
=NOTE=    you must first allocate or define a DD name HFMODPP.
=NOTE=
=NOTE=    This name points to a data set that contains the
=NOTE=    environment variables that are required by the ODPP
=NOTE=    to run as a TSO or batch application.
=NOTE=
=NOTE=
=NOTE=
=NOTE=    Where the data set member ENVVARS contains the following lines:
=NOTE=
=NOTE=    LIBPATH=/usr/local/odpp/odppbin
=NOTE=    ODPPLL=/usr/local/odpp/licensefiles
=NOTE=    ODPPERRL=/usr//local/odpp/odppbin
=NOTE=    ODPPTRCL=/usr/local/odpp/tracefiles
=NOTE=    ODPPTRC=N
=NOTE=
000008 //SYSIN DD *
=NOTE=
=NOTE=    DBP (Db2 Print)
=NOTE=
=NOTE=    Purpose
=NOTE=      To print the contents of a Db2. object in a variety of formats.
=NOTE=
=NOTE=
=NOTE=    Usage
=NOTE=      The Print utility function can:
=NOTE=        - Print data in table or single formats, in either character or
=NOTE=          HEX.
=NOTE=        - Select the rows and columns to be printed.
=NOTE=        - Limit the number of rows to be printed.
=NOTE=
⋮
```

Notes are identified by the characters `=NOTE=` in the line command field. The notes contain information to help in modifying the JCL statements and parameters.

5. Issue the RESET command to permanently remove the note lines, leaving only the data lines.
6. Overtype line 000008 with the A line command and then enter the MODEL primary command.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       EXAMPLE.JCL(NEWDBPX) - 01.00                  Columns 00001 00072
Command ===> model                                       Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=sysid,SQID=sqlid')
000002 //SYSPRINT DD SYSOUT=*
000003 //HFMTSPRT  DD SYSOUT=*
000004 //SYSOUT   DD SYSOUT=*
000005 //SYSTERM  DD SYSOUT=*
000006 //SYSTSPRT DD SYSOUT=*
000007 //HFMODPP DD DISP=SHR,DSN=OPTIM.ODPP.SAMPLIB(ENVVARS)
A      //SYSIN DD *
000009 /*
****** *************************** Bottom of Data ***************************
```

7. Enter DBP.

   The edit model for the ZDT/Db2 Print function is copied into the edit session.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       EXAMPLE.JCL(NEWDBPX) - 01.00                  Columns 00001 00072
Command ===>                                             Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=sysid,SQID=sqlid')
000002 //SYSPRINT DD SYSOUT=*
000003 //HFMTSPRT  DD SYSOUT=*
000004 //SYSOUT   DD SYSOUT=*
000005 //SYSTERM  DD SYSOUT=*
000006 //SYSTSPRT DD SYSOUT=*
000007 //HFMODPP DD DISP=SHR,DSN=OPTIM.ODPP.SAMPLIB(ENVVARS)
000008 //SYSIN DD *
=NOTE=
=NOTE=    DBP (Db2 Print)
=NOTE=
=NOTE=    Purpose
=NOTE=      To print the contents of a Db2. object in a variety of formats.
=NOTE=
=NOTE=
=NOTE=    Usage
=NOTE=      The Print utility function can:
=NOTE=        - Print data in table or single formats, in either character or
⋮
```

8. Once again, issue the RESET command to remove the note lines.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       EXAMPLE.JCL(NEWDBPX) - 01.00                  Columns 00001 00072
Command ===>                                             Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=sysid,SQID=sqlid')
000002 //SYSPRINT DD SYSOUT=*
000003 //HFMTSPRT  DD SYSOUT=*
000004 //SYSOUT   DD SYSOUT=*
000005 //SYSTERM  DD SYSOUT=*
000006 //SYSTSPRT DD SYSOUT=*
000007 //HFMODPP DD DISP=SHR,DSN=OPTIM.ODPP.SAMPLIB(ENVVARS)
000008 //SYSIN DD *
000009 $$FILEM DBP,
000010 $$FILEM FORMAT=TABLE|SINGLE,
000011 $$FILEM OBJIN=location.owner.name,
000012 $$FILEM OBJILOCN=location,
000013 $$FILEM OBJIOWNR=owner,
000014 $$FILEM OBJINAME=name,
000015 $$FILEM TINPUT=ddname,
000016 $$FILEM TINMEM=member,
000017 $$FILEM TMIN=template_data_set_name(member),
000018 $$FILEM UCREAD=YES|NO,
⋮
```

You now have a JCL member that can be used to print a Db2 table using the ZDT/Db2 DBP function.

Chapter 15. Interfacing with Db2® Interactive

To interface to Db2® Interactive (Db2I), use the Db2I utility function (option 5).

To display the Db2I Primary Option Menu panel, from the Primary Option Menu panel select option 5 (Db2I).

For further information about Db2I, see the *Db2® Application Programming and SQL Guide*.

Related references

Primary Option Menu panel on page 760
Db2I Primary Option Menu panel on page 635

Part II. ZDT/Db2 reference

Chapter 16. Db2® data types

ZDT/Db2 supports the following IBM-supplied data types:

Character strings (alphanumeric)

A sequence of bytes where the length of the string is the number of bytes in the sequence.

CHAR

Fixed-length character string.

VARCHAR

Varying-length character string.

GRAPHIC

Fixed-length graphic string.

VARGRAPHIC

Varying-length graphic string.

BINARY

Fixed-length binary string.

VARBINARY

Varying-length binary string.

Numeric

Decimal, binary integer, and floating point data types containing a sign and a precision.

DECIMAL or NUMERIC

A packed decimal number with an implied decimal point.

DECFLOAT

A decimal floating-point number.

SMALLINT

A binary integer with a precision of 15 bits.

INTEGER

A binary integer with a precision of 31 bits.

BIGINT

A binary integer with a precision of 63 bits.

REAL

A single precision floating-point number (32 bits).

DOUBLE, DOUBLE PRECISION, or FLOAT

A double precision floating-point number (64 bits).

Row identifier

Supports queries that navigate directly to a row in the table.

ROWID

A unique value maintained permanently by Db2®.

Datetime

These data types represent a point in time.

DATE

Internally to Db2®, the date is represented as a 4 byte field, with each byte containing 2 packed de cimal digits stored in the format *yyyymmdd*.

Externally, as returned by an SQL SELECT statement, the date is represented by a character field. The format of the string is dependant on the DATE parameter in the Db2® DSNHDECP installation module (the Db2® installation option specified on the installation panel DSNTIP4). Table 25: Date and Time field formats on page 443 shows valid Db2® date formats. The length of the date field is 10 bytes, except where the Db2® Installation Date format is LOCAL, in which case the length is installation-dependent and in the range 10 to 254 bytes. The date value must be in the range 0001-01-01 to 9999-12-31 (ISO format).

TIME

Internally to Db2® the timestamp is represented as a 7-15 byte field. with each byte containing 2 packed decimal digits. The first 4 bytes represent the date, the next 3 bytes the time, the next 0-6 bytes the microseconds, and when a TIME ZONE is present, 2 bytes containing the time zone.

Externally, as returned by an SQL select statement, the timestamp is represented by a 19-38 byte character field. The timestamp is always returned in the format yyyy-mm-dd.hh.mm.ss(.nnnnnnnnnnnn)(sHH:MM) where:

yyyy

An integer in the range 1 to 9999 representing the years.

mm

An integer in the range 1 to 12 representing the months.

dd

An integer in the range 1 to 31 representing the days. The value must be appropriate for the month, including leap years.

hh

An integer in the range 0 to 24 representing the hours.

mm

An integer in the range 0 to 59 representing the minutes.

ss

An integer in the range 0 to 59 representing the seconds.

The remaining parts of the TIMESTAMP (fractional part of the seconds value and TIME ZONE) are optional:

nnnnnnnnnnnn

An integer in the range 0 to 999999999999 representing the fractional part of the seconds. The number of digits can vary from 0-12 inclusive, depending on the precision of the TIMESTAMP.

sHH

A signed integer in the range -12 to +14 inclusive. -12 represents 12 hours behind Greenwich mean time, +14 represents 14 hours ahead.

MM

An integer in the range 00-59, representing the minutes component of the time zone. Internally to Db2®, the time is represented as a 3 byte field with each byte containing 2 packed decimal digits stored in the format *hhmmss*.

Table 25. Date and Time field formats

Format Name	Abbreviation	Date format	Time format
International Standard Organization	ISO	yyyy-mm-dd	hh.mm.ss
IBM® USA standard	USA	mm/dd/yyyy	hh.mm AM or PM
IBM® European standard	EUR	dd.mm.yyyy	hh.mm.ss
Japanese industrial standard	JIS	yyyy-mm-dd	hh:mm:ss

Table 25. Date and Time field formats (continued)

Format Name	Abbreviation	Date format	Time format
Installation defined	LOCAL	Installation-defined format	Installation-defined format

For a full description of Db2® data types, see the *Db2 UDB for z/OS SQL Reference*.

Large objects (LOBs)

The term *large object (LOB)* refers to any of the following data types:

CLOB

Character large object

DBCLOB

Double-byte character large object

BLOB

Binary large object

ZDT/Db2 recognizes columns with a LOB data type but these columns do not participate in ZDT/Db2 functions. If displayed, the name of the LOB column is shown but the column is displayed as empty.

Chapter 17. ZDT/Db2 panels and fields

This section of the manual lists most of the ZDT/Db2 panels. For several of the panels, there is a definition for each field in the panel and, where applicable, the value ranges that are valid for each entry field.

Scrollable fields on ZDT/Db2 panels

Some fields on ZDT/Db2 panels are scrollable to allow you to input or view longer values.

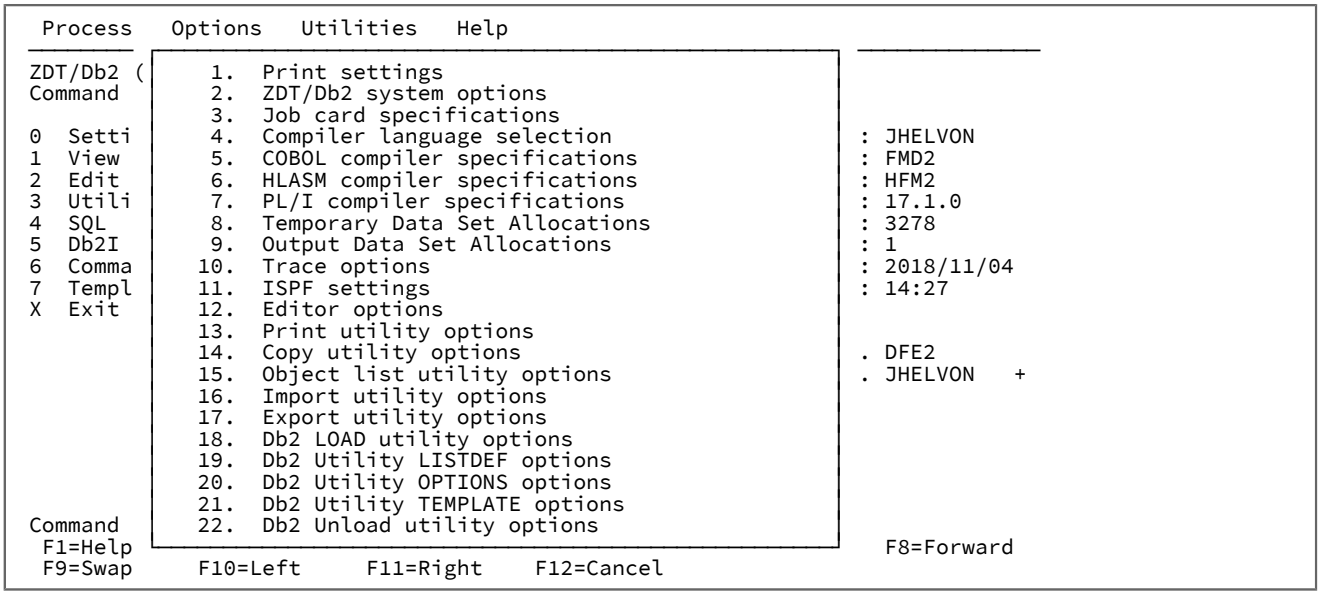
The panels shown in this section were produced using Db2® Version 8. In these panels, some fields have adjacent "**+**" and "**-**" signs. These signs show that the fields are scrollable.

For further information, see Scrollable input and display fields for long names on page 40.

Action bar pull-down menu

The Action bar pull-down menus give you a fast way to move around the product. The panel shown here shows an example of a pull-down menu for the action bar item, **Options**.

Panel and field definitions



Process

The Process menu offers the following choice:

1

Exit

Options

Note: The Options menu is not available for the Set Processing Options panel (Set Processing Options panel on page 832).

The Options menu offers the following choice:

1
Print settings

2
ZDT/Db2 system options

3
Job card specifications

4
Compiler language selection

5
COBOL compiler specifications

6
HLASM compiler specifications

7
PL/I compiler specifications

8
Temporary Data Set Allocations

9
Output Data Set Allocations

10
Trace options

11
ISPF settings

12
Editor options

13
Print utility options

14

Copy utility options

15

Object list utility options

16

Import utility options

17

Export utility options

18

Db2® LOAD utility options

19

Db2® Utility LISTDEF options

20

Db2® Utility OPTIONS options

21

Db2® Utility TEMPLATE options

22

Db2® Unload utility options

Utilities

Note: The Utilities menu is not available for the Set Processing Options panel (Set Processing Options panel on page 832).

The Utilities menu offers the following choice:

1

Print

2

Objects

3

Copy

4

Object List

5

Db2® Privileges

6

Import

7

Export

8

Create

9

Db2® Utilities

10

Audit Trail

11

Print Browse

Help

The Help menu offers the following choice:

1

Help for help...

2

Extended help...

3

Keys help...

4

Help index A-M

5

Help index N-Z

6

Tutorial...

7

About...

8

About Db2®...

9

News about ZDT/Db2...

Parent panels

Pull-down menus are available from all ZDT/Db2 panels.

Child panels

None.

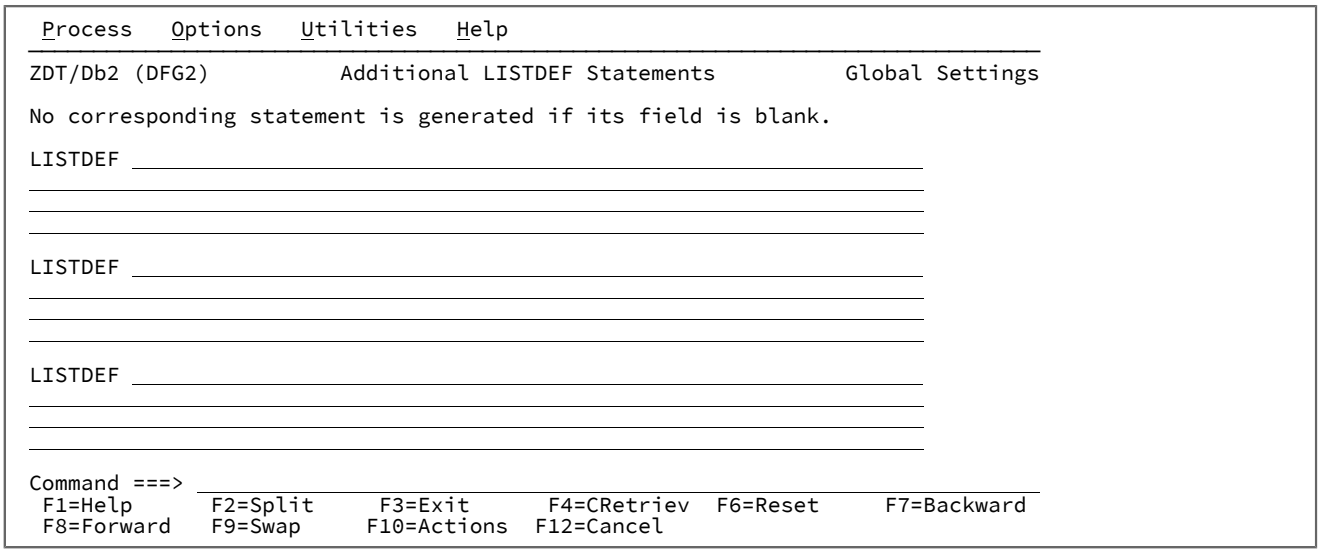
Related tasks

- Checking your ZDT/Db2 version on page 27
- Exiting from ZDT/Db2 on page 29
- Selecting an item from the action bar on page 40

Additional LISTDEF Statements panel

You use the Additional LISTDEF Statements panel to provide additional LISTDEF statements, which are used to supplement the LISTDEF statement provided in the LISTDEF Options panel.

Panel and field definitions



LISTDEF

The text that you enter here is included in a LISTDEF statement, if the Use additional statements option on the LISTDEF Options panel (LISTDEF Options panel on page 729) is selected.

Parent panels

- LISTDEF Options panel on page 729

Child panels

None.

Related tasks

- RESET primary command on page 994
- Setting options for Db2 utilities on page 348

Additional TEMPLATE Statements panel

You use the Additional TEMPLATE Statements panel to provide additional TEMPLATE statements, which are used to supplement the TEMPLATE statement provided in the TEMPLATE Options (1 of 2) panel.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)         Additional TEMPLATE Statements          Global Settings

No corresponding statement is generated if its field is blank.

TEMPLATE ____________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________

TEMPLATE ____________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________

TEMPLATE ____________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________

Command ===> _______________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

TEMPLATE

The text that you enter here is included in a TEMPLATE statement, if the Use additional statements option on the TEMPLATE Options (1 of 2) panel (TEMPLATE Options (1 of 2) panel on page 897) is selected.

Parent panels

- TEMPLATE Options (1 of 2) panel on page 897

Child panels

None.

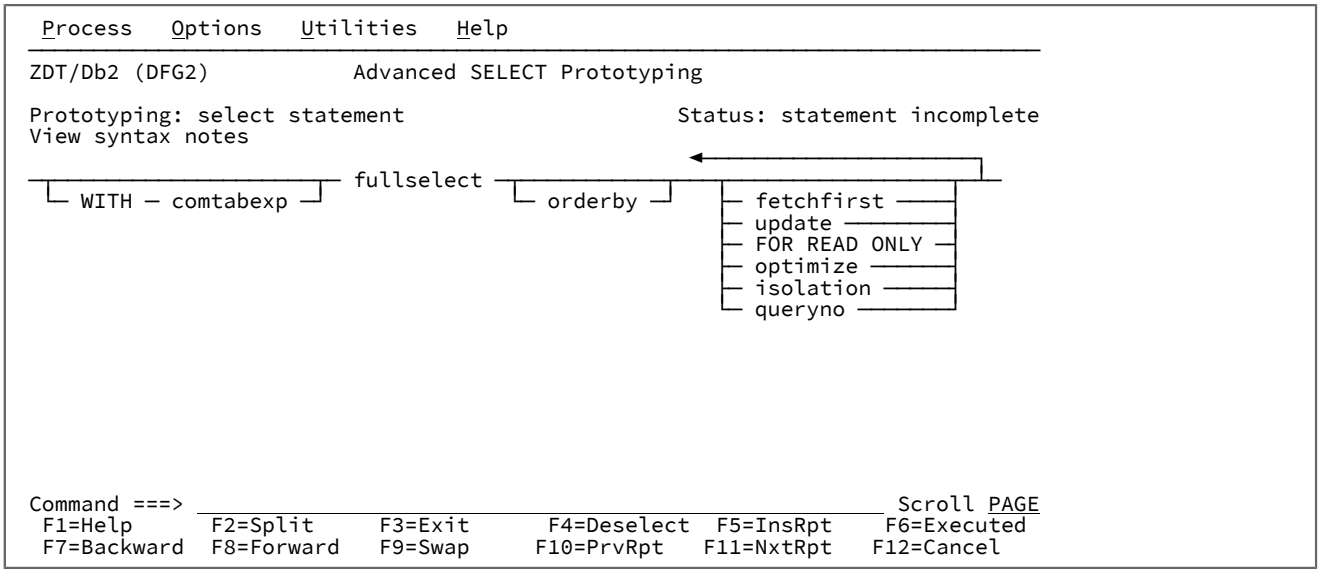
Related tasks

- RESET primary command on page 994
- Setting options for Db2 utilities on page 348

Advanced SELECT Prototyping panel

The advanced select prototyping function is an aid to learning the syntax of the SQL SELECT statement, and a tool for prototyping complex SQL statements. Fragments of the statement syntax are displayed on the screen in a form based on the "railroad tracks" used to describe the syntax in the Db2® manuals. By positioning the cursor on the parts of the syntax you wish to explore or use, you can build up a statement that conforms to the syntax requirements. The statement you create is displayed at the bottom of the screen, and can be executed when you have completed it.

Panel and field definitions



Prototyping

Shows the name of the syntax fragment currently displayed.

Status

The current status of the SQL statement you are building. The status indicator, in the form of a text string, applies either to the whole SQL statement (in which case, it uses the word *"statement"*), or to the clause within the SQL statement that you are currently prototyping (in which case, it uses the word *"clause"*):

statement incomplete

Only is displayed on the first Advanced SELECT Prototyping panel, and indicates that the statement generated so far is not ready for execution.

clause incomplete

Can be displayed on any panel except the first, and indicates that the clause currently being prototyped is not ready for execution (and by implication, the statement as a whole is not yet ready for execution).

clause complete

Can be displayed on any panel except the first, and indicates that the clause currently being prototyped is now complete. This indirectly implies (because otherwise the status would appear as *"statement complete"*) that the statement as a whole is not yet ready for execution.

statement complete

Can be displayed on any panel, and indicates that the statement as a whole is now complete and ready for execution (and implies that the current clause is also complete).

Note: Advanced SELECT prototyping is designed to help you build a SELECT statement until it is displayed to be complete. However, the *"completed"* status of a SELECT clause or statement does not necessarily mean that it is syntactically correct or references valid Db2® data. Only the successful execution of the SELECT statement confirms that its syntax is correct.

View syntax notes

Explanatory notes relating to the current syntax fragment, available on some of the advanced SELECT prototyping panels. Such panels show the text *"View syntax notes"* immediately below the *"Prototyping:"* field heading.

To view the notes, move the cursor to the *"View syntax notes"* text and press Enter. ZDT/Db2 displays a panel containing notes relevant to the current syntax fragment.

Parent panels

- SQL Prototyping, Execution and Analysis panel on page 843

Child panels

To display this panel…	Use/do this
Select Statement Browse panel on page 815	Press Enter

Related tasks

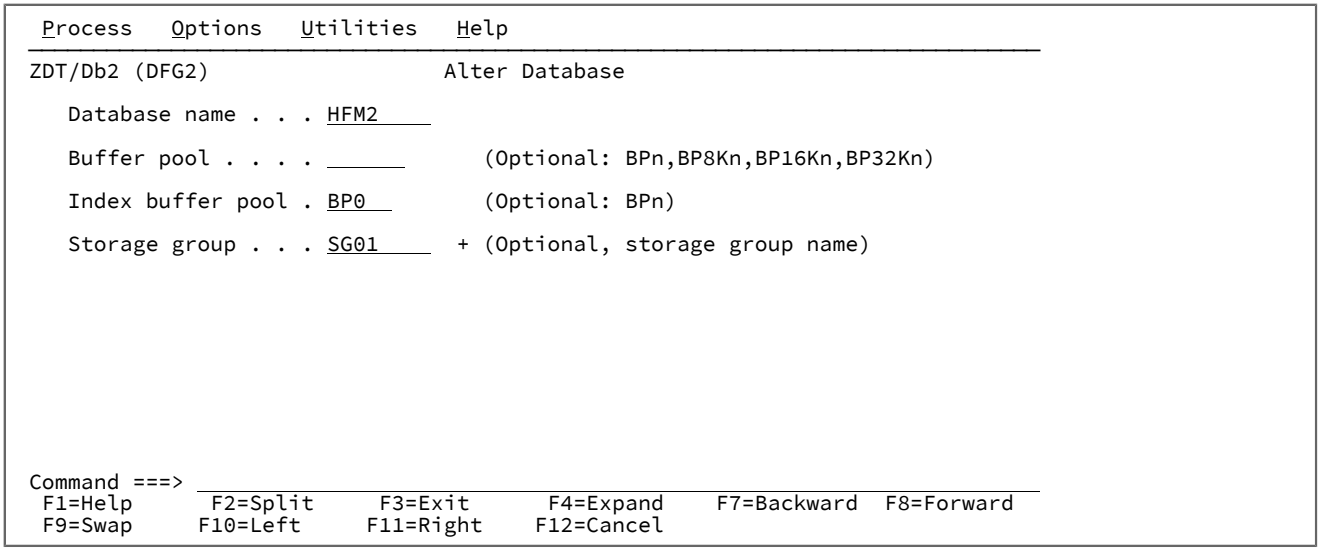
- Using advanced SQL prototyping on page 399
- Generated SELECT statement color-coding on page 401

Related references

- CANCEL primary command on page 926
- DESELECT primary command on page 938
- EXECUTE primary command on page 948
- INSRPT primary command on page 971
- NEXTRPT primary command on page 984
- PREVRPT primary command on page 988
- SQL primary command on page 1005

Alter Database panel

Panel and field definitions



Parent panels

- Databases panel on page 609

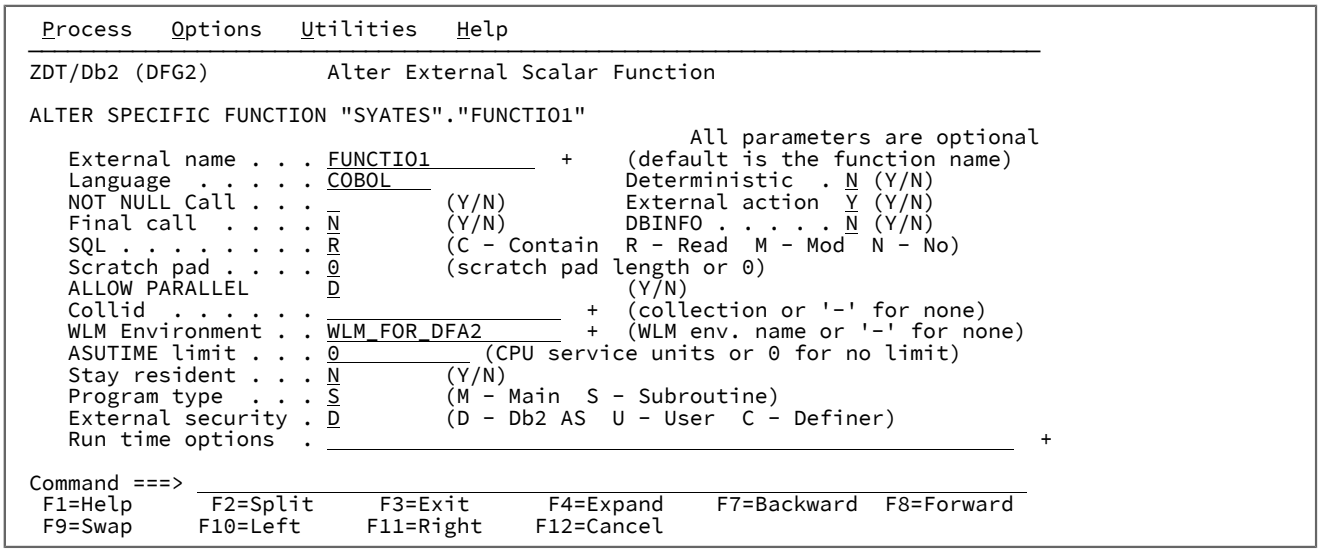
Child panels

None.

Related tasks

Alter External Scalar Function panel

Panel and field definitions



Parent panels

- Functions panel on page 709

Child panels

None.

Related tasks

Alter Index panel

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                  Alter Index

   Index owner . . . . HFMUSER          +
   Index name  . . . . XDEPT1                                            +

   Partition . . . . . _________

   Primary quantity  . _________         Buffer pool  . . . . BP0
   Secondary quantity  _________         Erase rule . . . . . NO
   Free page . . . . . _________         Close rule . . . . . NO
   Percent free  . . . _________
   Storage group . . . _________  +
   VSAM catalog  . . . _________         Piece size(KB) . . . 2097152
   GB pool cache . . . _________         Copy allowed . . . . NO
   Limit value . . . . _____________________________________________________  +

Use the SHOW primary command to display index partition values, and the SAVE
line command to copy the values to this panel.

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Parent panels

- Indexes panel on page 723

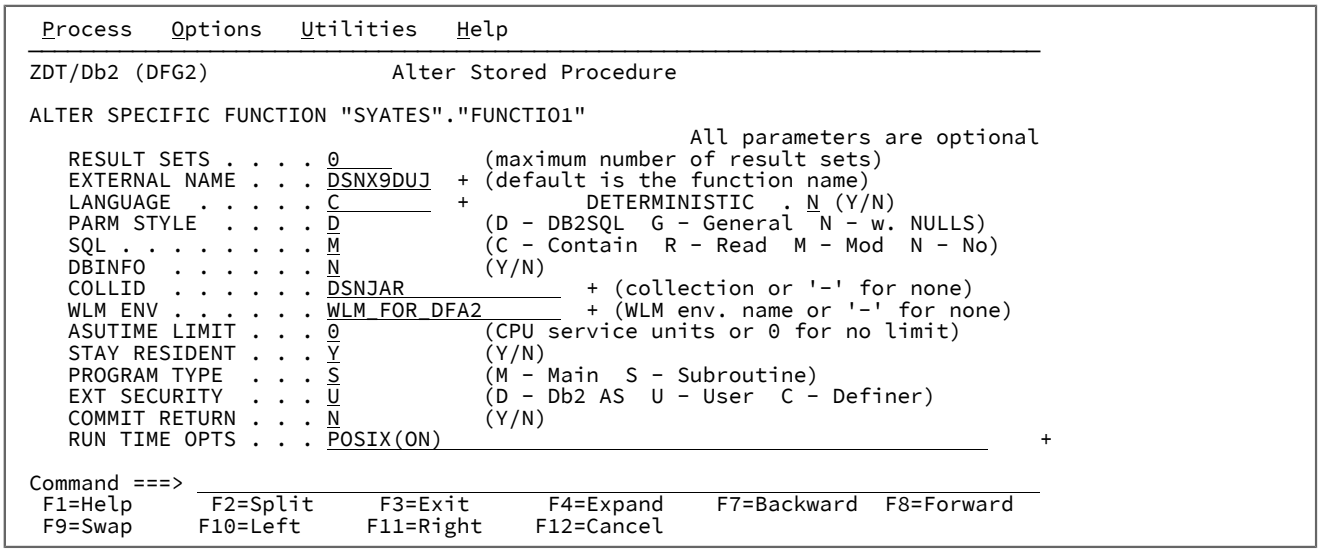
Child panels

None.

Related tasks

Alter Stored Procedure panel

Panel and field definitions



Parent panels

- Stored Procedures panel on page 852

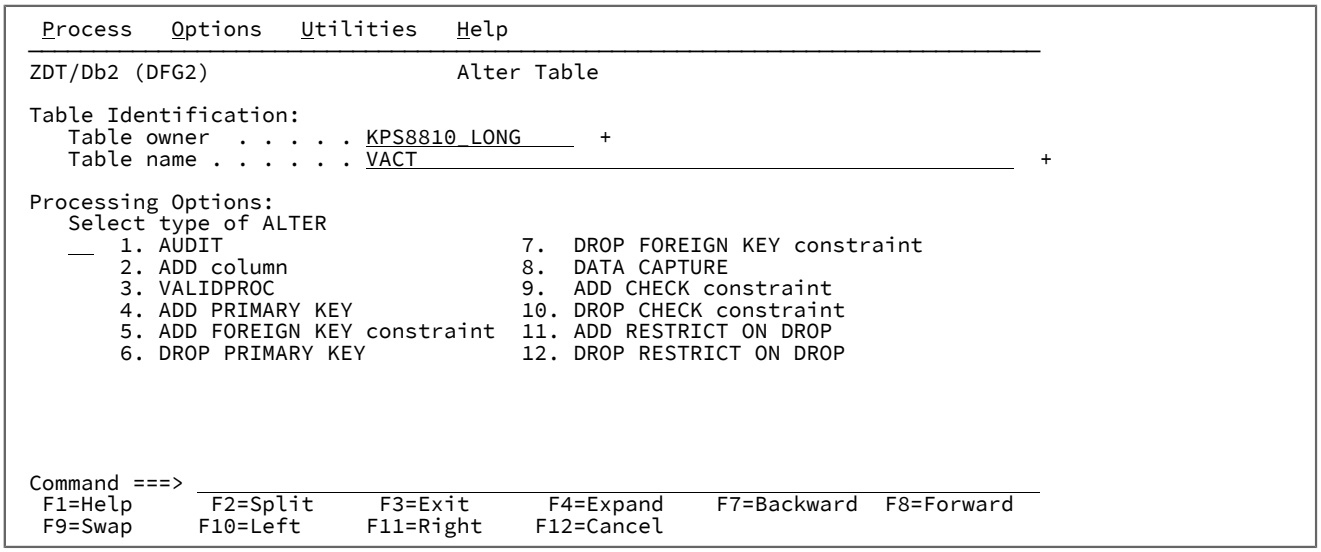
Child panels

None.

Related tasks

Alter Table panel

Panel and field definitions



Parent panels

- Tables, Views and Aliases panel on page 891
- Table Check Constraints panel on page 868

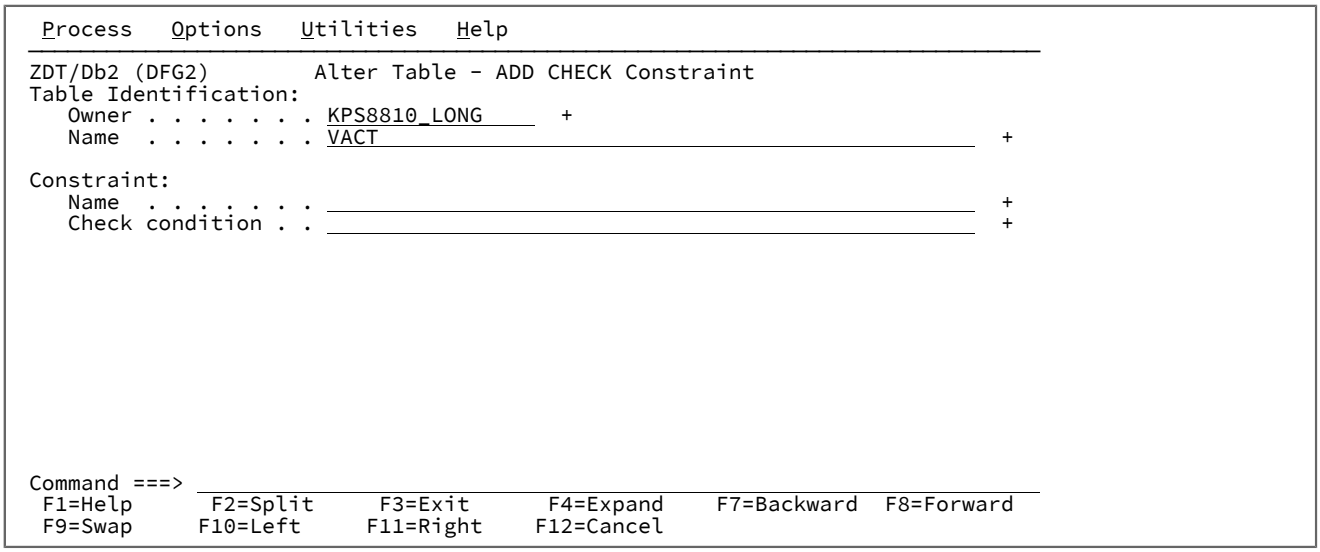
Child panels

To display this panel...	Use/do this
Alter Table - AUDIT panel on page 462	Processing option 1
Alter Table - ADD Column panel on page 459	Processing option 2
Alter Table - VALIDPROC panel on page 468	Processing option 3
Alter Table - ADD PRIMARY KEY panel on page 461	Processing option 4
Alter Table - ADD FOREIGN KEY Constraint panel on page 460	Processing option 5
Alter Table - DROP PRIMARY KEY panel on page 466	Processing option 6
Alter Table - DROP FOREIGN KEY Constraint panel on page 465	Processing option 7
Alter Table - DATA CAPTURE panel on page 463	Processing option 8
Alter Table - ADD CHECK Constraint panel on page 458	Processing option 9
Alter Table - DROP CHECK Constraint panel on page 464	Processing option 10
Alter Table - RESTRICT ON DROP panel on page 467	Processing option 11 or 12

Related tasks

Alter Table - ADD CHECK Constraint panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

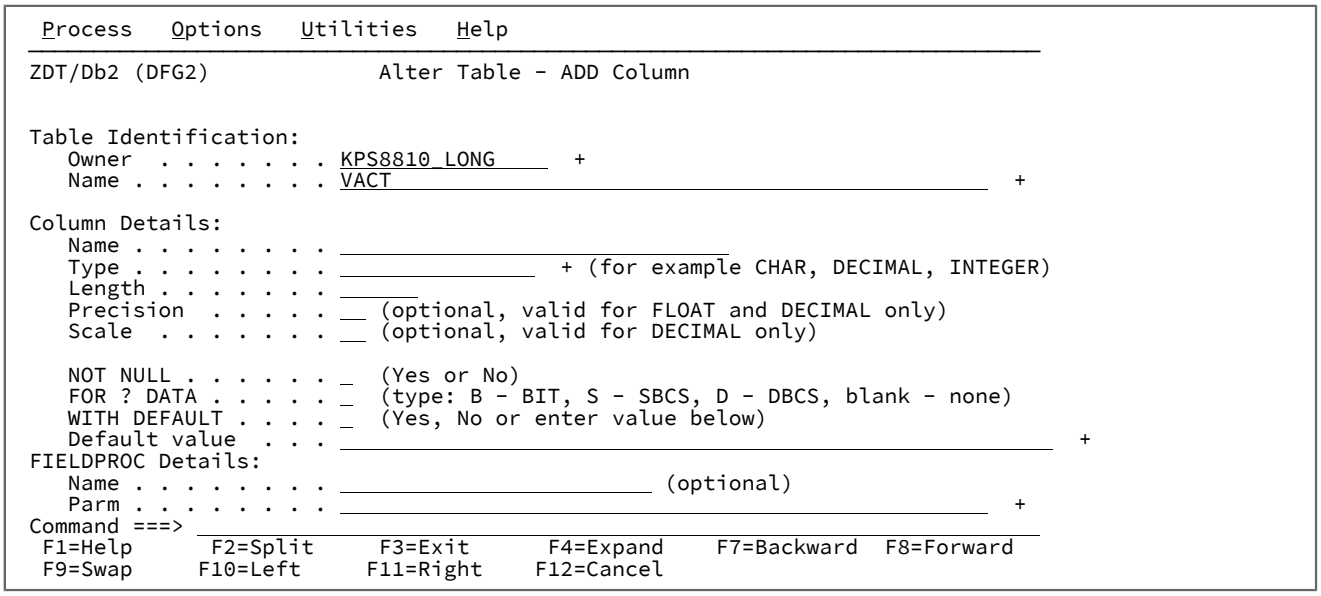
Child panels

None.

Related tasks

Alter Table - ADD Column panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

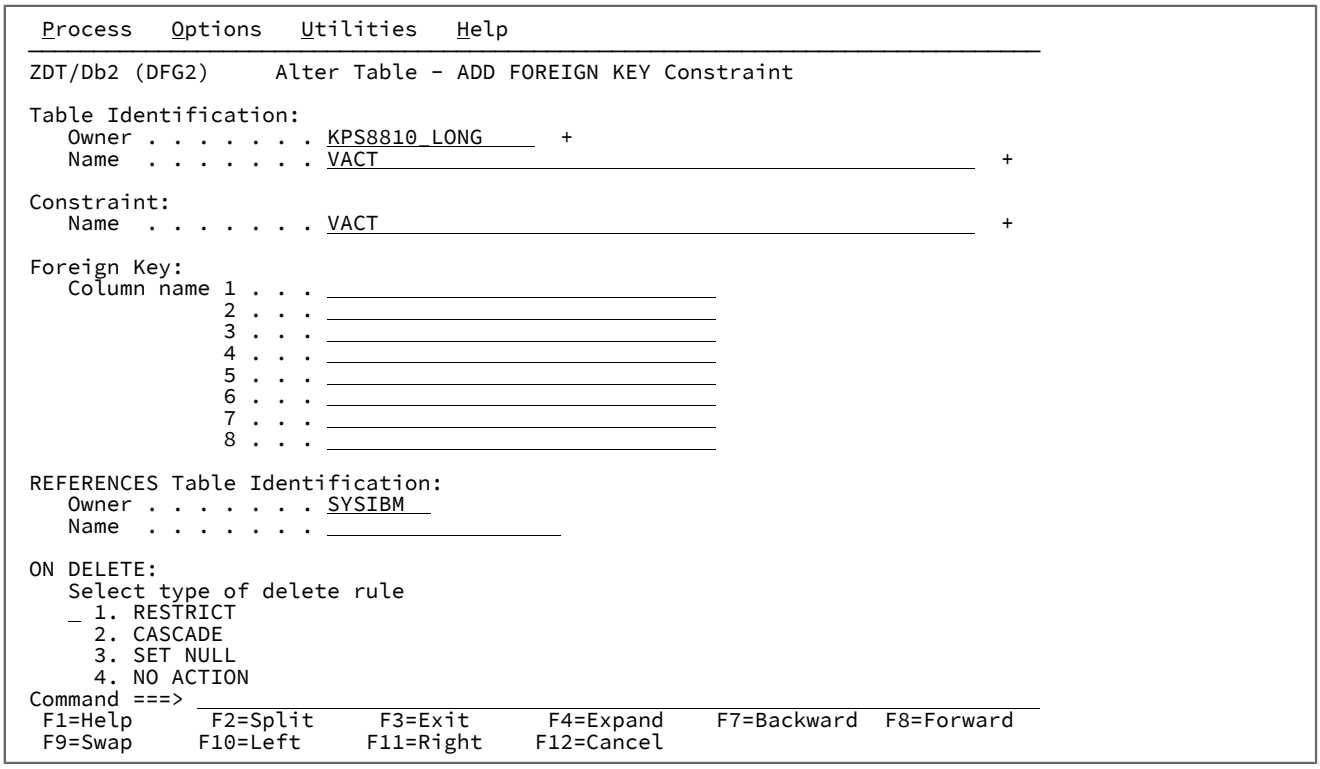
Child panels

None.

Related tasks

Alter Table - ADD FOREIGN KEY Constraint panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

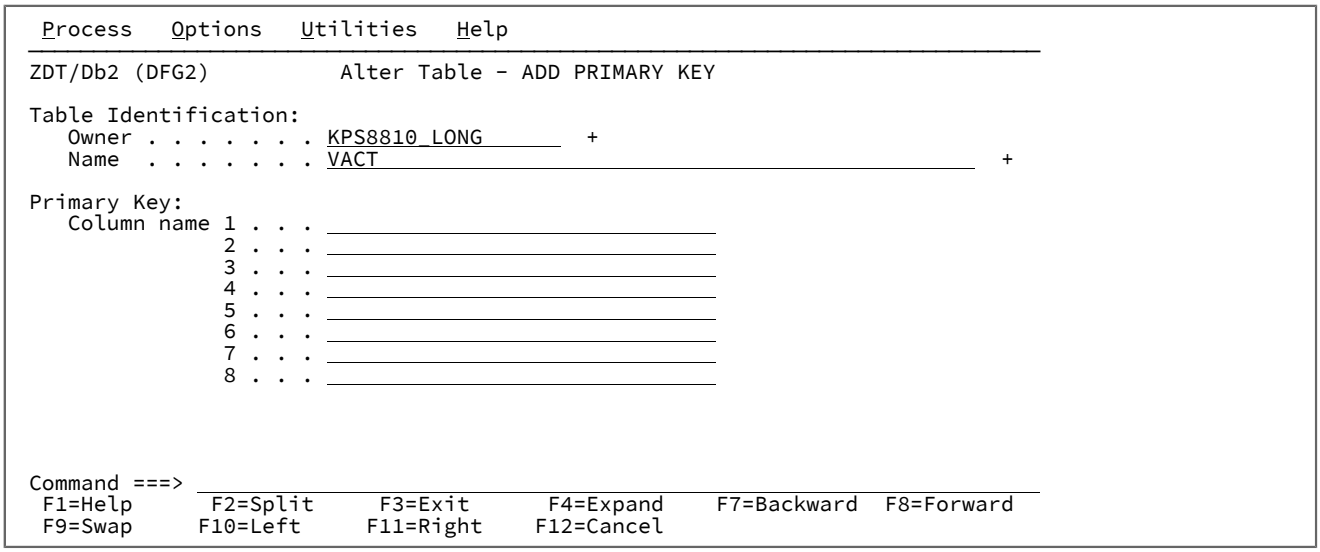
Child panels

None.

Related tasks

Alter Table - ADD PRIMARY KEY panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

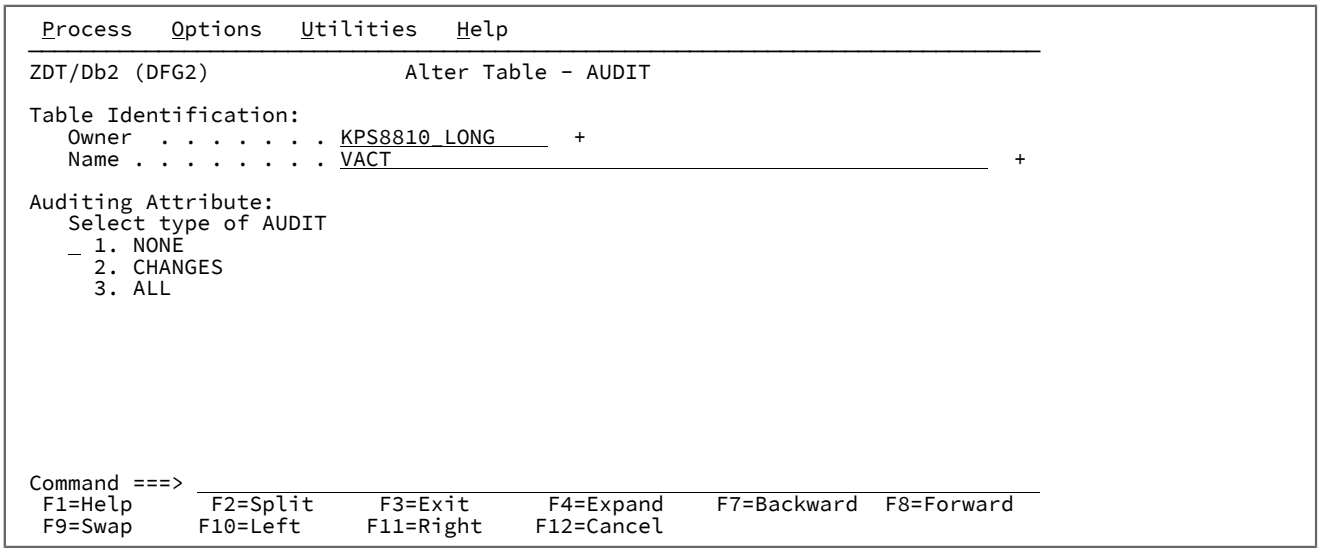
Child panels

None.

Related tasks

Alter Table - AUDIT panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

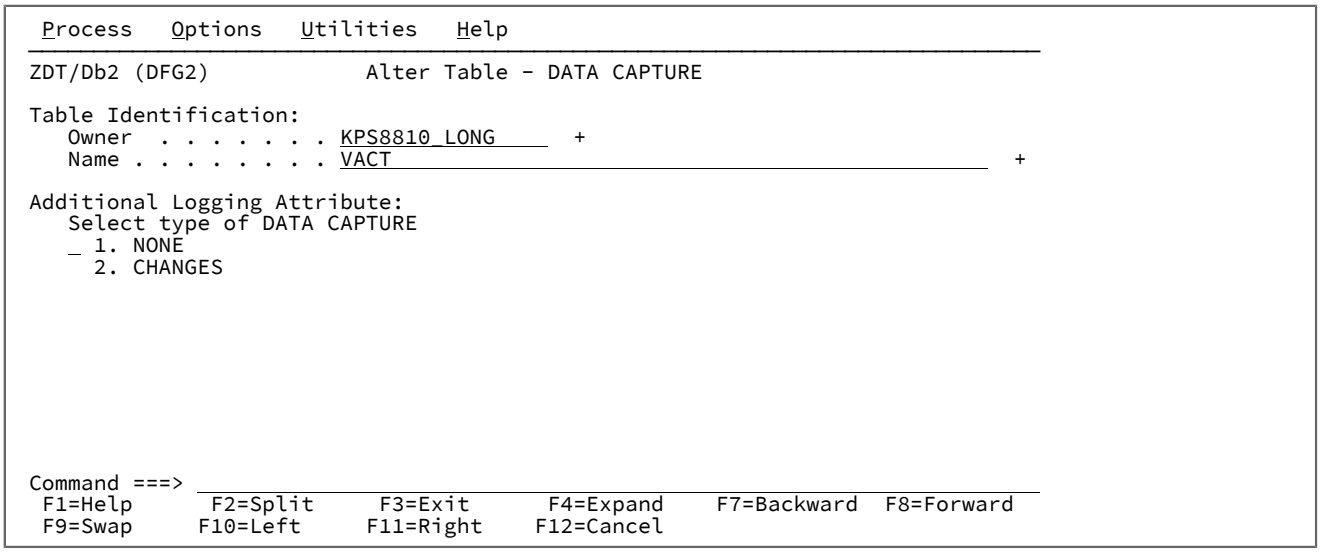
Child panels

None.

Related tasks

Alter Table - DATA CAPTURE panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

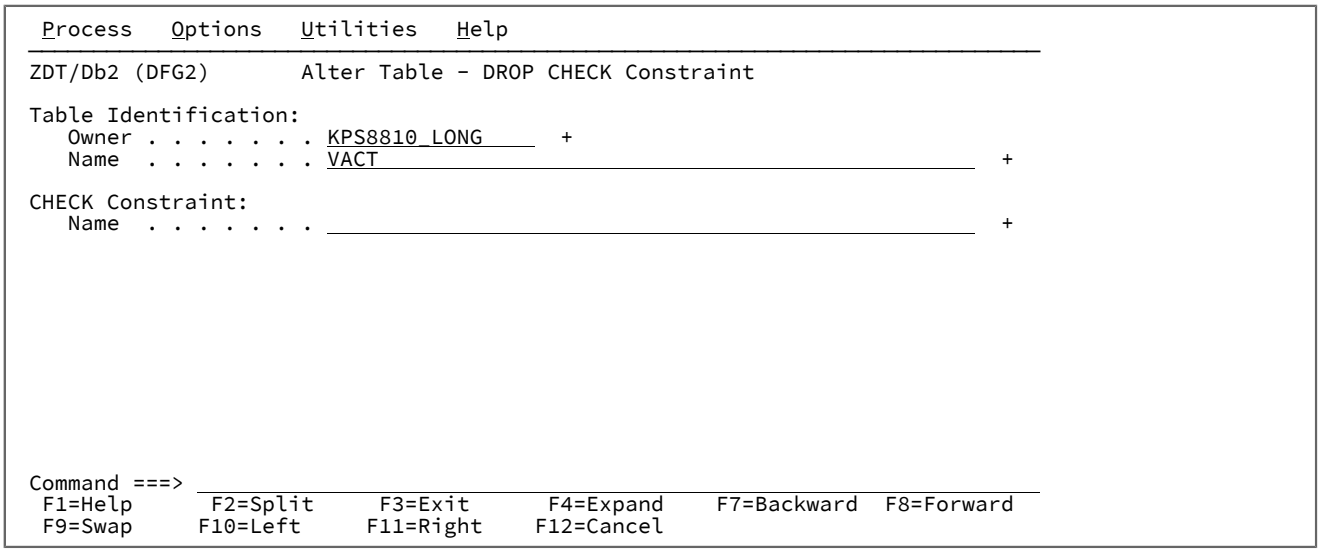
Child panels

None.

Related tasks

Alter Table - DROP CHECK Constraint panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457
- Table Check Constraints panel on page 868

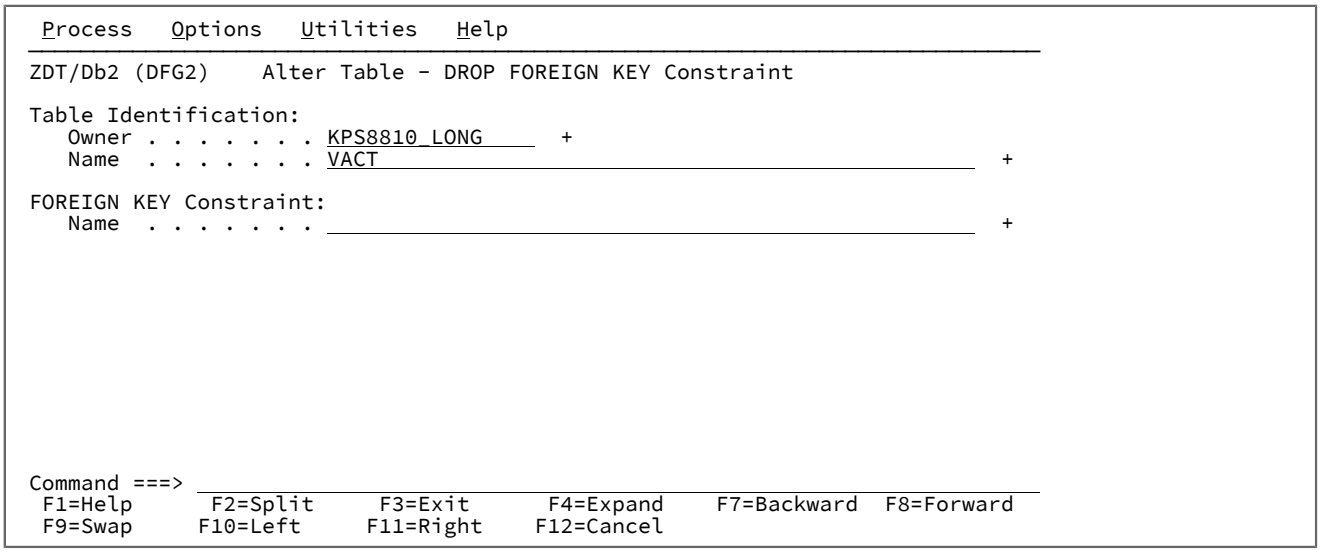
Child panels

None.

Related tasks

Alter Table - DROP FOREIGN KEY Constraint panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

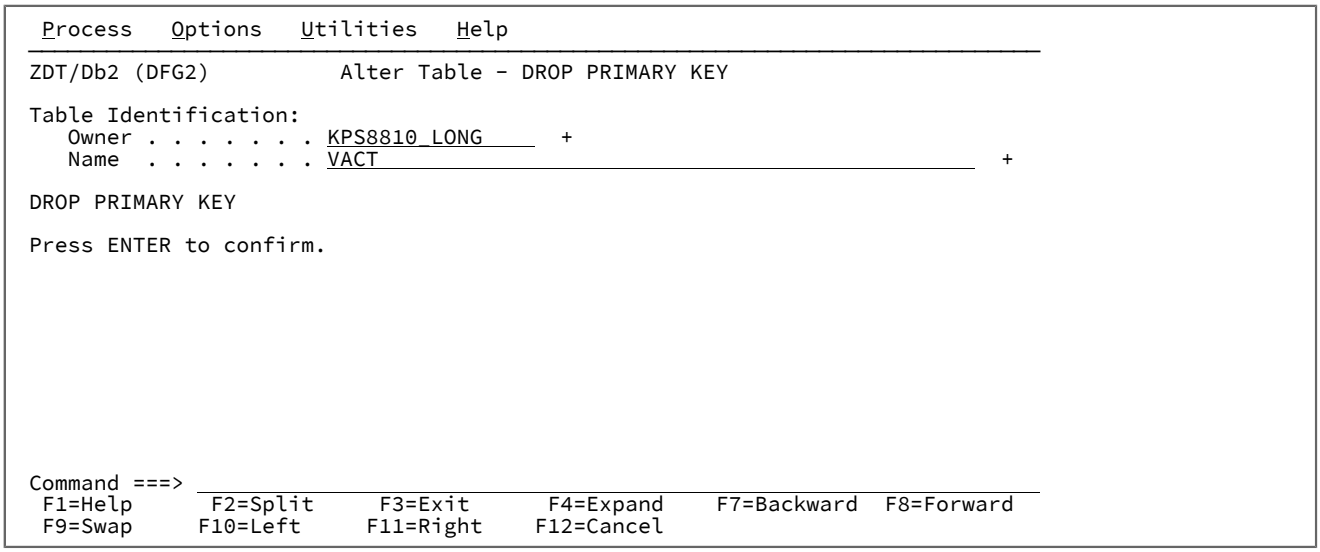
Child panels

None.

Related tasks

Alter Table - DROP PRIMARY KEY panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

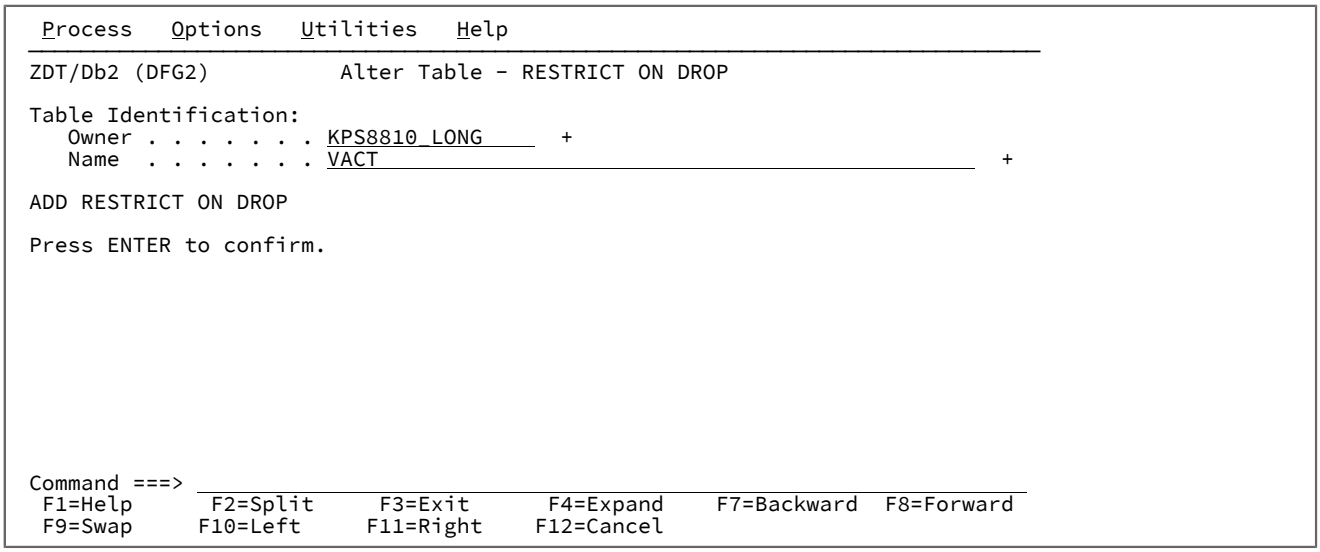
Child panels

None.

Related tasks

Alter Table - RESTRICT ON DROP panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

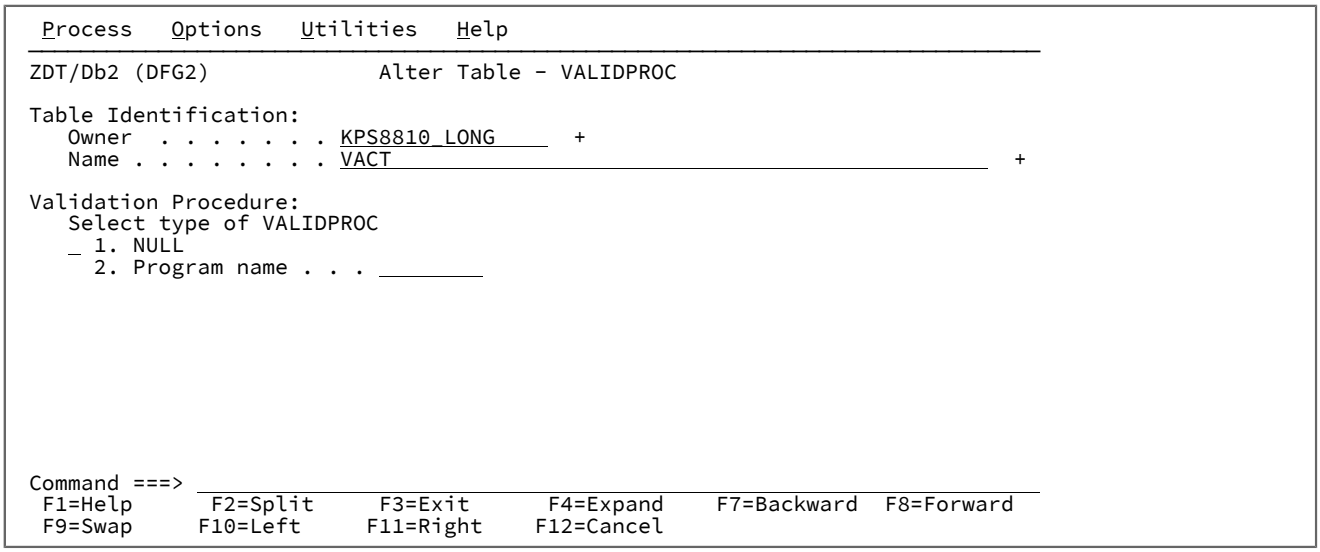
Child panels

None.

Related tasks

Alter Table - VALIDPROC panel

Panel and field definitions



Parent panels

- Alter Table panel on page 457

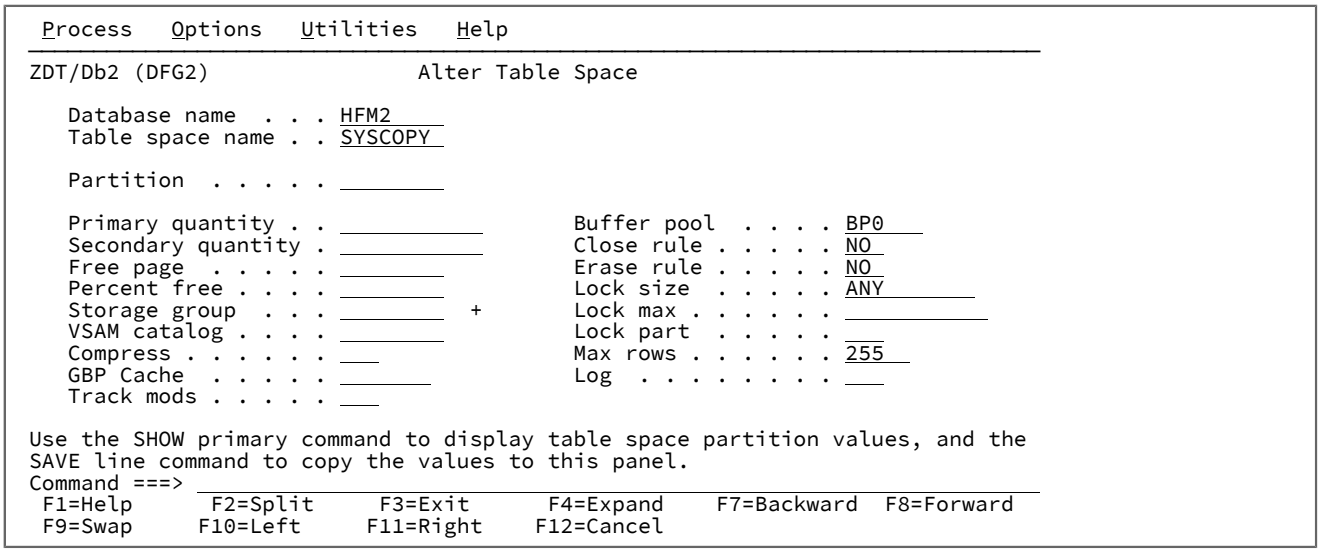
Child panels

None.

Related tasks

Alter Table Space panel

Panel and field definitions



Parent panels

- Table Spaces panel on page 879

Child panels

None.

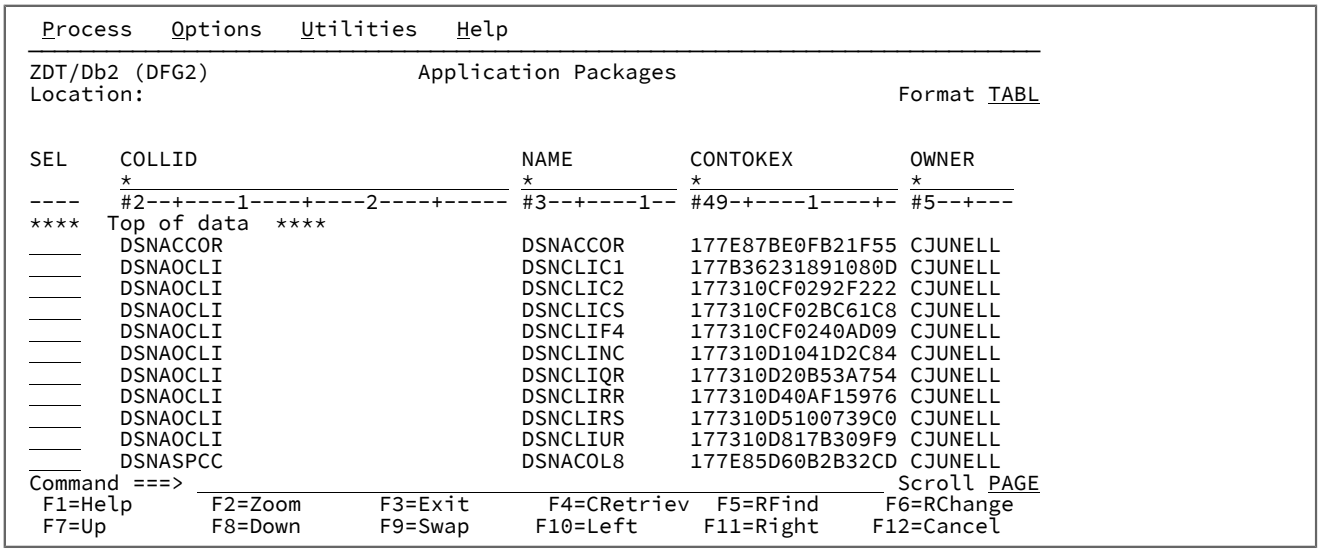
Related tasks

Application Packages panel

You use the Application Packages panel to list packages in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSPACKAGE.



Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Tables, Views and Aliases panel on page 891
- Indexes panel on page 723
- Application Plans panel on page 473
- Collections panel on page 482
- Stored Procedures panel on page 852
- Triggers panel on page 903

- Plan Table Rows panel on page 754
- Statement Table Rows panel on page 845

Child panels

To display this panel...	Use/do this
Bind Package panel on page 482	Line command BI or CBI
Package Dependencies panel on page 753	Line command DEP
Free Package panel on page 708	Line command FRE
Grant privileges panels on page 715	Line command G
Details panels on page 636	Line command I
Plan Table Rows panel on page 754	Line command LPT
Privileges panels on page 768	Line command P
Package List panel on page 753	Line command PKL
Application Plans panel on page 473	Line command PL
Revoke privileges panels on page 798	Line command R
	Line command RBI

To display this panel...	Use/do this
Rebind Package panel on page 772	
	Line command ROW
Display Row panel on page 637	
	Line command S
Table Spaces panel on page 879	
	Line command SQL
(ISPF panel)	
Tables, Views and Aliases panel on page 891	Line command TB
	Line command V
Tables, Views and Aliases panel on page 891	
	Line command VER
Application Packages panel on page 469	
	Line command X
Indexes panel on page 723	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Application Plans panel

You use the Application Plans panel to list application plans in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSPLAN.

```
  Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                Application Plans
Location:                                                    Format TABL


SEL    NAME     CREATOR  VALIDAEX  ISOLATEX          VALIDEX OPERATEX
       *        *        *         *                 *       *
----   #1--+--- #2--+--- #44-+---- #45-+----1----+- #46-+-- #47-+---
****  Top of data  ****
_____  DSNACLI  CJUNELL  Run time  Repeatable read  Yes     Yes
_____  DSNEDCL  CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNESPCS CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNESPRR CJUNELL  Run time  Repeatable read  Yes     Yes
_____  DSNHYCRD CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNREXX  CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNTEP81 CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNTIA81 CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNTP481 CJUNELL  Run time  Cursor stability Yes     Yes
_____  DSNWZP   CJUNELL  Run time  Cursor stability Yes     Yes
_____  HFM2GEN  SHRIKES  Bind time Cursor stability Yes     Yes
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Application Packages panel on page 469
- Collections panel on page 482
- Database Request Modules panel on page 607

Child panels

To display this panel…	Use/do this
Bind Application Plan panel on page 481	Line command BI
Free Application Plan panel on page 708	Line command FRE
Grant privileges panels on page 715	Line command G
Details panels on page 636	Line command I
Plan Table Rows panel on page 754	Line command LPT
Database Request Modules panel on page 607	Line command M
Privileges panels on page 768	Line command P
Application Packages panel on page 469	Line command PKG
Package List panel on page 753	Line command PKL

To display this panel…	Use/do this
	Line command R
Revoke privileges panels on page 798	
	Line command RBI
Rebind Application Plan panel on page 770	
	Line command ROW
Display Row panel on page 637	
	Line command S
Table Spaces panel on page 879	
	Line command SQL
(ISPF panel)	
	Line command T
Tables, Views and Aliases panel on page 891	
	Line command V
Tables, Views and Aliases panel on page 891	
	Line command X
Indexes panel on page 723	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Basic SELECT Prototyping panel

You use the Basic SELECT Prototyping (primary) panel to specify up to 15 tables to be used to construct a FROM clause as the starting point for prototyping a SELECT statement.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)             Basic SELECT Prototyping

Enter the name(s) of the table(s) from which to retrieve data:
    Owner           Name
 #1 DSN8810       + EMP                            + Location ________________
 #2 DSN8810       + EEMP                           + Database _________
 #3 ____________  + ________________________________ + Tbl spc. _________
 #4 ____________  + ________________________________ +
 #5 ____________  + ________________________________ +
 #6 ____________  + ________________________________ +
 #7 ____________  + ________________________________ +
 #8 ____________  + ________________________________ +
 #9 ____________  + ________________________________ +
#10 ____________  + ________________________________ +
#11 ____________  + ________________________________ +
#12 ____________  + ________________________________ +
#13 ____________  + ________________________________ +
#14 ____________  + ________________________________ +
#15 ____________  + ________________________________ +
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Owner

This field is optional. If specified, it is used to qualify the name field on the same line. If omitted, the current SQL ID is used.

Name

This field is required. It specifies the name of the table to be processed.

Location

This field is optional. If specified, it is used to qualify all the tables entered in the **Owner** and **Name** fields. If omitted, the current location is be used.

Database

This field is optional. If specified, all the tables entered in the **Owner** and **Name** fields must exist in the given database. If omitted, the tables can exist in any database at the specified or defaulted location.

Table space

This field is optional. If specified, all the tables entered in the **Owner** and **Name** fields must exist in the given table space. If omitted, the tables can exist in any table space at the specified or defaulted location, subject to any restriction imposed by the database specification.

After you have specified the table names on the primary Basic SELECT Prototyping panel, press Enter to display the secondary Basic SELECT Prototyping panel.

```
Process   Options   Utilities   Help
-------------------------------------------------------------------------------
 ZDT/Db2 (DFG2)             Basic SELECT Prototyping               Row 1 of 19

 SELECT ?
   FROM ?
   WHERE ?
   ORDER BY ?

 Row count  ALL          Number of rows to display

 Select columns (S/A/D) or enter predicates to build the SELECT statement:

 S  LOp ( Tab Column Name   Data Type(length)    Op Value                     )
 _ ___ _  #1 EMPNO          CHAR(6)              __ ___________________________ _
 _ ___ _  #1 FIRSTNME       VARCHAR(12)          __ ___________________________ _
 _ ___ _  #1 MIDINIT        CHAR(1)              __ ___________________________ _
 _ ___ _  #1 LASTNAME       VARCHAR(15)          __ ___________________________ _
 _ ___ _  #1 WORKDEPT       CHAR(3)              __ ___________________________ _
 _ ___ _  #1 PHONENO        CHAR(4)              __ ___________________________ _
 _ ___ _  #1 HIREDATE       DATE(4)              __ ___________________________ _
 _ ___ _  #1 JOB            CHAR(8)              __ ___________________________ _
 _ ___ _  #1 EDLEVEL        SMALLINT(2)          __ ___________________________ _
 _ ___ _  #1 SEX            CHAR(1)              __ ___________________________ _
 _ ___ _  #1 BIRTHDATE      DATE(4)              __ ___________________________ _
 _ ___ _  #1 SALARY         DECIMAL(9,2)         __ ___________________________ _
 _ ___ _  #1 BONUS          DECIMAL(9,2)         __ ___________________________ _
 _ ___ _  #1 COMM           DECIMAL(9,2)         __ ___________________________ _
 _ ___ _  #2 DEPTNO         CHAR(3)              __ ___________________________ _
 _ ___ _  #2 DEPTNAME       VARCHAR(36)          __ ___________________________ _
 _ ___ _  #2 MGRNO          CHAR(6)              __ ___________________________ _
 _ ___ _  #2 ADMRDEPT       CHAR(3)              __ ___________________________ _
 _ ___ _  #2 LOCATION       CHAR(16)             __ ___________________________ _
          ****  End of data  ****

 Command ===> _____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=Expand    F6=Execute   F7=Backward
  F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Row count

Enter a number in this field to control:

- The number of rows returned in the result set when the SQL statement is executed. Specifying 0 results in all qualifying rows being returned. Specifying *n* (when *n*>0) limits the number of rows returned to the lesser of: *n* rows, or the number of rows in the result table.
- Whether or not an "OPTIMIZE FOR *n* ROWS" clause is added to the SQL statement prior to execution. Specifying *n* (when *n*>0) results in the addition of "OPTIMIZE FOR *n* ROWS" to the SQL statement prior to execution. Specifying a 0 value does not add "OPTIMIZE FOR *n* ROWS" to the SQL statement. Specifying a non-zero value may improve Db2's use of resources when executing the SQL statement.

S

A selection column in which you can specify one of the following values for the column on that line:

S

Select. Adds the column name to the column list in the SELECT clause.

A

Select ascending. Adds the column name to the column list in the SELECT clause, and adds the column name with the keyword `ASC` to the column list in the ORDER BY clause.

D

Select descending. Adds the column name to the column list in the SELECT clause, and adds the column name with the keyword `DESC` to the column list in the ORDER BY clause.

LOp

Logical operator to join a new predicate to previous predicates. You can specify one of the following values for the column on that line:

AND (the default operator)
OR
NOT (equivalent to *"and not"*)
NOR (equivalent to *"or not"*)

(

Inserts an opening parenthesis at the start of the predicate added to the WHERE clause. Use an opening parenthesis in conjunction with a closing parenthesis to ensure the predicate for this column is evaluated before other (non-parenthesized) predicates.

Tab

The correlation name of the table containing the column on this line. This is an information-only field.

Column Name

The name of the column as held in the Db2® catalog. This is an information-only field.

Data Type(length)

The Db2® data type of the column and its length, or numeric precision and scale. This is an information-only field.

Op

Expression operator. You can specify one of the following values for the column on that line:

=

Equal. This is the default operator.

<>

Not equal.

>

Greater than.

<

Less than.

>=

Greater than or equal to.

<=

Less than or equal to.

IN

In *set*. If the value in the column is in the specified set of values (specified in **Value** field), the result of the expression is true. The IN expression operator generates the predicate IN *set* in the WHERE clause.

NI

Not in *set*. If the value in the column is not in the specified set of values (specified in **Value** field), the result of the expression is true. The NI expression operator generates the predicate NOT IN *set* in the WHERE clause.

LI

Like *pattern*. If the value in the column matches the pattern (specified in **Value** field), the result of the expression is true. The LI expression operator generates the predicate LIKE *pattern* in the WHERE clause.

NL

Not like *pattern*. If the value in the column does not match the pattern (specified in **Value** field), the result of the expression is true. The NL expression operator generates the predicate NOT LIKE *pattern* in the WHERE clause.

Value

The value to be used with the expression operator (specified in the **Op** field).

The format of the data you enter depends on the expression operator:

- For expression operators =, <>, >, <, >=, and <=:
  - For columns with an alphanumeric data type, specify a character string, optionally enclosed in quotation marks. If you do not enclose the string in quotation marks, ZDT/Db2 automatically adds the quotation marks around the string when it adds the generated predicate to the WHEN clause.

    Examples:

    ```
    SMITH
    'SMITH'
    SALES REP
    'SALES REP'
    ```

  - For columns with a numeric data type, specify a numeric value.

Examples:

```
500
33.75
0
```

- You can also specify a column name as a value. The column name must be qualified with its correlation name.

  Examples:

  ```
  #1.WORKDEPT
  #2.BONUS
  ```

  Note: To enter a string that begins with a "#", you must enclose the string in quotation marks.

- For expression operators IN and NI:
  - For columns with an alphanumeric data type, specify a list of character strings. Each character string *must* be enclosed in quotation marks and separated by a comma.

    Examples:

    ```
    'SMITH','JONES','BROWN'
    'ANALYST',   'DESIGNER', 'OPERATOR','MANAGER'
    ```

  - For columns with a numeric data type, specify a list of numeric values. Each value must be separated by a comma and, optionally, one or more spaces.

    Examples:

    ```
    101,102,103
    45.5, 50.0, 65.5
    ```

  - You can also specify a list of column names. Each column name must be qualified with its correlation name and be separated by a comma. Examples:

    ```
    #1.ACCT,#1PROD,#2MAINT
    #1.BONUS,   #2.BONUS
    ```

- For the expression operator LI:
  - Specify a pattern (as a character string), optionally enclosed in quotation marks.

    Note: You can only specify a pattern for columns with an alphanumeric data type.

    If you do not enclose the pattern in quotation marks, ZDT/Db2 automatically adds the quotation marks around the pattern when it adds the generated predicate to the WHEN clause.

    The pattern can contain an underscore (_) to represent any single character, or a percent sign (%) to represent a string of zero or more characters.

    All of the following examples find the string "SMITH":

```
'SMITH'
SMI
'SM%'
SMI__
```

- Specify a column name as a value. The column name must be qualified with its correlation name.

  Examples:

  ```
  #1.SEX
  #1.WORKDEPT
  ```

)

Inserts a closing parenthesis at the end of the predicate added to the WHERE clause. Use in conjunction with an opening parenthesis to ensure the predicate for this column is evaluated before other (non-parenthesized) predicates.

Parent panels

- Tables, Views and Aliases panel on page 891
- SQL Prototyping, Execution and Analysis panel on page 843

Child panels

To display this panel…	Use/do this
Select Statement Browse panel on page 815	Press Enter

Related tasks

- Using basic SQL prototyping on page 390

Related references

- ALL primary command on page 922
- CANCEL primary command on page 926
- EXECUTE primary command on page 948
- RESET primary command on page 994
- SQL primary command on page 1005
- UNDO primary command on page 1008

Bind Application Plan panel

ZDT/Db2 displays the Bind Application Plan panel if you use the BI line command against an item listed on the Application Plans panel on page 473.

Bind Package panel

ZDT/Db2 displays the Bind Package panel if you use either the BI or CBI line command against an item listed on the Application Packages panel on page 469.

Children of Table panel

ZDT/Db2 displays the Children of Table panel if you use the CHR line command against an item listed on the Tables, Views and Aliases panel on page 891.

Collections panel

You use the Collections panel list to list collections in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSPACKAGES.

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                    Collections
Location:                                                    Format TABL

                              PKG
SEL    COLLID                COUNT
       *                     ____*
----   #2--+----1----+---  ----#4
****  Top of data  ****
_____  DSNACCOR                 1
_____  DSNAOCLI                 9
_____  DSNASPCC                12
_____  DSNEDCL                  1
_____  DSNESPCS                 1
_____  DSNESPRR                 1
_____  DSNESPRR$$$$$$$$$$       1
_____  DSNHYCRD                 1
_____  DSNJAR                   8
_____  DSNREXCS                 1
_____  DSNREXRR                 1
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748

Child panels

To display this panel…	Use/do this
	Line command G
Grant privileges panels on page 715	
	Line command P
Privileges panels on page 768	
	Line command PKG
Application Packages panel on page 469	
	Line command PKL
Package List panel on page 753	
	Line command PL
Application Plans panel on page 473	
	Line command R

To display this panel…	Use/do this
Revoke privileges panels on page 798	
	Line command ROW
Display Row panel on page 637	
	Line command SQL
(ISPF panel)	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

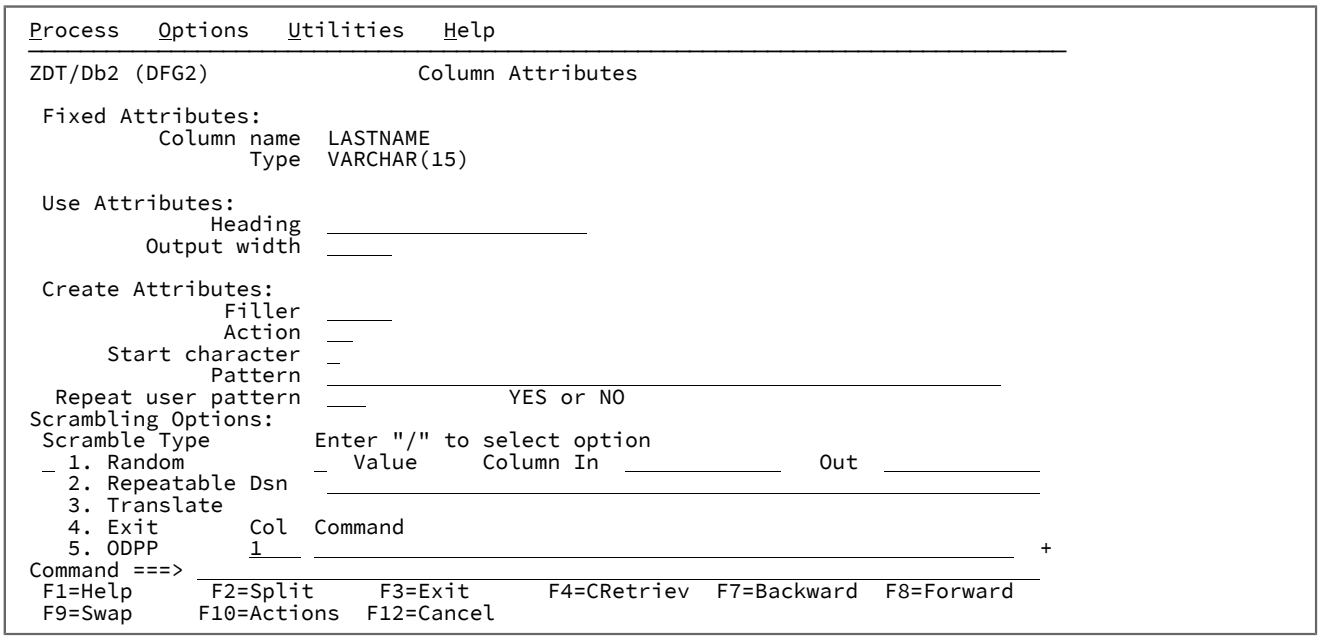
Related references

- SORT primary command on page 1003

Column Attributes panel (alphanumeric)

The Column Attributes (alphanumeric) panel describes the formatting and data create attributes for an alphanumeric column.

Panel and field definitions



Column name

The name of the column.

Type

The Db2® data type of the column.

Heading

The alternative column heading.

Output width

The number of character positions used by view, edit, browse, and print functions to show the column in TABL display or print format.

Filler

The value placed in each byte of the column before any other operation in the construction of the column. The default value is a blank character.

char

This character, such as 0, is written in each byte.

X'cc'

This hexadecimal value, such as X'FF', is written in each byte.

Action

This column sets how the contents of the column are varied for each row. The default action is FX.

FX

Contents of the column remain *fixed*.

RA

Contents of the column are generated from characters selected at *random* from the specified pattern. A different set of characters is selected for each row.

RO

Contents of the column *rolled*. The specified pattern is shifted left one byte for each row until the first nonblank character of the pattern is in the first byte of the column. Then the pattern is shifted right one byte for each output row until it returns to its original position in the column. RO can only be used with a user-supplied pattern, not with an HCL-supplied pattern (see following description for **Pattern**).

RP

Contents of the column are *rippled*. The specified pattern is shifted left one byte for each row and the truncated character is appended to the right end of the pattern.

SL

Contents of the column are *shifted left* for each row. The specified pattern is shifted left one character and filled to the right with a space. When all characters in the pattern have been shifted out, the original pattern is restored and the process is repeated.

SR

Contents of the column are *shifted right* for each row. The specified pattern is shifted right one character and filled to the left with a space. When all characters in the pattern have been shifted out, the original pattern is restored and the process is repeated.

TL

Contents of the column are *truncated* on the *left* for each row. The specified pattern is left-truncated (the leftmost character replaced with a space) one character at a time until all characters in the pattern have been truncated. Then the original pattern is restored and the process is repeated.

TR

Contents of the column are *truncated* on the *right* for each row. The specified pattern is right-truncated (the rightmost character replaced with a space) one character at a time until all characters in the pattern have been truncated. Then the original pattern is restored and the process is repeated.

WV

Contents of the column are *waved*. The specified pattern is shifted left one byte for each row until the first nonblank character of the pattern is in the first byte of the column. Then the original pattern is restored and the process repeated. WV can only be used with a user-supplied pattern, not with an HCL-supplied pattern (see following description for **Pattern**).

Start character

Starting character used when you specify an HCL-supplied pattern (AL, AN, or CO). The specified character must be one of the characters in the HCL-supplied pattern. The default start character is the first character in the HCL-supplied pattern.

Pattern

The pattern used when generating data for this column. You can specify either an HCL-supplied pattern indicator or a quoted string containing a user-supplied pattern. The HCL-supplied patterns you can specify are:

AL

Alphabetic. Characters A–Z inclusive.

AN

Alphanumeric. Characters A–Z, 0–9, inclusive.

CO

Collating sequence. All printable characters in the range X'40'–X'F9'.

If you provide a user-supplied pattern that is longer than the column, for actions other than RA and RP, the pattern is first truncated on the right to fit the receiving column before the specified action is performed. For the RA action, characters are randomly selected from the entire user-supplied pattern. For the RP action, the entire pattern is rippled for each row before it is truncated to fit the receiving column.

If you provide a user-supplied pattern that is shorter than the column, you can specify that you want the pattern repeated to fill the column.

Repeat user pattern

Specify YES if you have provided a user-supplied pattern that is shorter than the receiving column. This option has no effect when the RA action is specified. The default value is NO.

NO

Fill character used to pad the receiving column.

YES

Supplied pattern repeated as many times as necessary to fill the receiving column.

Note: HCL-supplied patterns are always repeated in a column, as necessary.

Scrambling Options

These options control column scrambling used during copy, import, or export functions to produce test data.

Scramble Type

Specify one of the following values:

Blank

No scrambling is performed. Value or range specifications are saved but ignored for the associated function.

1 (Random)

Performs random scrambling. The same input value produces different output values on subsequent invocations.

2 (Repeatable)

Performs repeatable scrambling. The same input value produces the same output value on subsequent invocations.

3 (Translate)

Performs translation. The value data set is searched to find a matching input value. If a match is found, then the output value is taken from the output column of the matching record.

4 (Exit)

Invokes a scrambling user exit. ZDT/Db2 displays a panel where you can specify the user exit name and user exit parameters and options.

5 (ODPP)

Performs data scrambling using the ODPP command specified in the adjacent command field. This is the Optim™ Data Privacy Provider **Command** input area. The **Col** field is the left column position, which can be used to position to the error column when a syntax error occurs. Refer to the ODPP Syntax Guide for the command syntax.

The ODPP command is a scrollable field. Use the F10 and F11 keys to scroll left and right. Use the F4 key to expand the command into a separate window. This option is only available if Z Data Tools has been customized to access ODPP, and the environment has been successfully initialised. For more information, see the *Z Data Tools Customization Guide*.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

Related tasks

- Editing the template on page 245
- Changing column attributes on page 104
- Changing the attributes for an alphanumeric column on page 105

Column Attributes panel (DATE)

The Column Attributes (DATE) panel describes the formatting and data create attributes for a DATE column.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                Column Attributes

Fixed Attributes:
         Column name  BIRTHDATE
               Type  DATE
             Format  dd.mm.yyyy

Use Attributes:
             Heading  ____________________
        Output width  _____

Create Attributes:
          Date value  ____________________________________________
                            (Use = for date now, * for runtime date)
            Increment  0_________



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Column name

The name of the column.

Type

The Db2® data type of the column.

Format

The format in which the date is displayed or printed.

Heading

The alternative column heading.

Output width

The number of character positions used by view, edit, and print functions to show the column in TABL display or print format.

Date value

The initial value a column is to contain. Valid values are:

start date

A date in a valid Db2® or local format.

*** (asterisk)**

The date the data is created. Not valid if the Db2® installation date format is LOCAL.

=

The current date. Not valid if the Db2® installation date format is LOCAL.

Increment

A integer that represents the number of days to add to the date each cycle.

The default increment value is 0.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

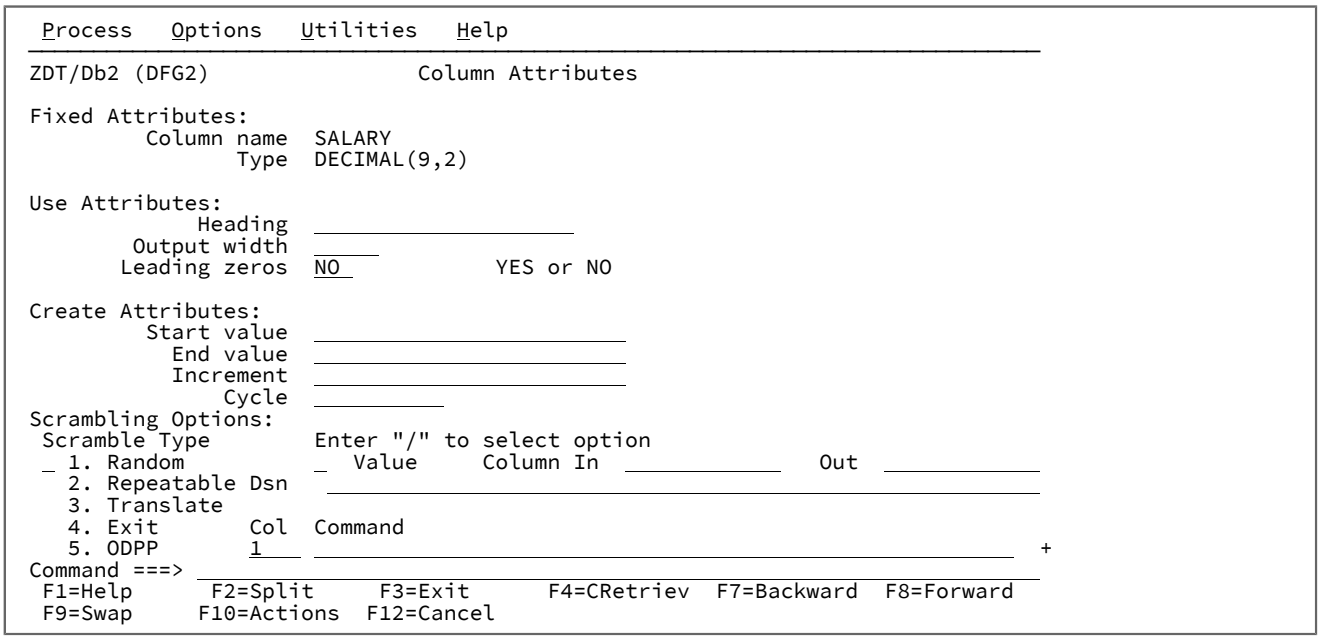
Related tasks

- Editing the template on page 245
- Changing column attributes on page 104
- Table 25: Date and Time field formats on page 443

Column Attributes panel (numeric)

The Column Attributes (numeric) panel describes the formatting and data create attributes for a numeric column.

Panel and field definitions



Column name

The name of the column.

Type

The Db2® data type of the column.

Heading

The alternative column heading.

Output width

The number of character positions used by view, edit, and print functions to show the column in TABL display or print format.

Leading zeros

Specifies whether leading zeros are to be shown when this field is displayed or printed in SNGL or TABL display or print format. Specify YES if you want leading zeros shown. Specify NO if you want leading zeros replaced with blanks.

Start value

The initial value a column is to contain, before being adjusted by the specified increment. The value must be a number that the column can hold without truncation or loss of data. The value is converted to the appropriate numeric data type if required. For a column containing decimal places, the value specified cannot contain more decimal places than specified by the column definition. If you specify a negative number, the sign is honored, even if the column is unsigned. If the column is a floating-point column, the start value can be specified as a floating-point number consisting of a mantissa and an exponent (such as -1.14579E01). The mantissa consists

of an optional sign (+ or -) followed by 1 to 16 digits. The mantissa can also contain a decimal point. The exponent consists of the letter E, an optional sign (+ or 0), and 1 or 2 digits.

The default start value is 0.

End value

The maximum value (if the increment is a positive number) or minimum value (if the increment is a negative number) a column is to contain. The value must be a number that the column can hold without truncation or loss of data. The value is converted to the appropriate numeric data type if required. For a column containing decimal places, the value specified cannot contain more decimal places than specified by the column definition. If you specify a negative number, the sign is honored, even if the column is unsigned. If the column is a floating-point column, the end value can be specified as a floating-point number consisting of a mantissa and an exponent (such as -1.14579E01). The mantissa consists of an optional sign (+ or -) followed by 1 to 16 digits. The mantissa can also contain a decimal point. The exponent consists of the letter E, an optional sign (+ or 0), and 1 or 2 digits.

The default end value is the largest positive or negative number the column can contain.

Increment

The positive or negative number by which the value in the column is adjusted for each row (or cycle of rows). The increment value must be a number that the column can hold without truncation or loss of data. The increment value is converted to the appropriate numeric data type if required. For a column containing decimal places, the value specified cannot contain more decimal places than specified by the column definition. If the column is a floating-point column, the increment can be specified as a floating-point number consisting of a mantissa and an exponent (such as -1.14579E01). The mantissa consists of an optional sign (+ or -) followed by 1 to 16 digits. The mantissa can also contain a decimal point. The exponent consists of the letter E, an optional sign (+ or 0), and 1 or 2 digits.

For the first row (or cycle of rows), the column is set to the start value you specify. For each subsequent row (or cycle of rows), the increment is added to the value in the column. This process continues while the calculated value in the column does not exceed the specified end value. That is, while the calculated value is less than or equal to the end value if the increment is a positive number, or greater than or equal to the end value if the increment is a negative number. When the end value is exceeded, the value in the column remains unchanged for all subsequent rows.

If the increment value is zero, the value in the column is always set to the start value.

The default increment is 0.

Cycle

The number of output rows that are generated before the increment value is applied to the column value. For example, if you specify a column start value of 100, an increment value of 10, and a cycle value of 3, the column in the first 3 rows contains 100, 110 in next 3 rows, 120 in the next 3 rows, and so on.

If cycle is zero, the value in the column is always set to the start value.

The default cycle is 1.

Scrambling Options

These options control column scrambling used during copy, import, or export functions to produce test data.

Scramble Type

Specify one of the following values:

Blank

No scrambling is performed. Value or range specifications are saved but ignored for the associated function.

1 (Random)

Performs random scrambling. The same input value produces different output values on subsequent invocations.

2 (Repeatable)

Performs repeatable scrambling. The same input value produces the same output value on subsequent invocations.

3 (Translate)

Performs translation. The value data set is searched to find a matching input value. If a match is found, then the output value is taken from the output column of the matching record.

4 (Exit)

Invokes a scrambling user exit. ZDT/Db2 displays a panel where you can specify the user exit name and user exit parameters and options.

5 (ODPP)

Performs data scrambling using the ODPP command specified in the adjacent command field. This is the Optim™ Data Privacy Provider **Command** input area. The **Col** field is the left column position, which can be used to position to the error column when a syntax error occurs. Refer to the ODPP Syntax Guide for the command syntax.

The ODPP command is a scrollable field. Use the F10 and F11 keys to scroll left and right. Use the F4 key to expand the command into a separate window. This option is only available if Z Data Tools has been customized to access ODPP, and the environment has been successfully initialised. For more information, see the *Z Data Tools Customization Guide*.



Note:
For numeric columns, setting **Leading zeros** to YES causes leading zeros to be scrambled when random or repeatable scrambling (**Value** option blank) has been requested.

For example:

- With **Leading zeros** set to YES, 00123 may produce 56872 (zeros changed).
- With **Leading zeros** set to NO, 00123 may produce 00343 (zeros unchanged).

For repeatable scrambling, setting **Leading zeros** to YES guarantees a unique result.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

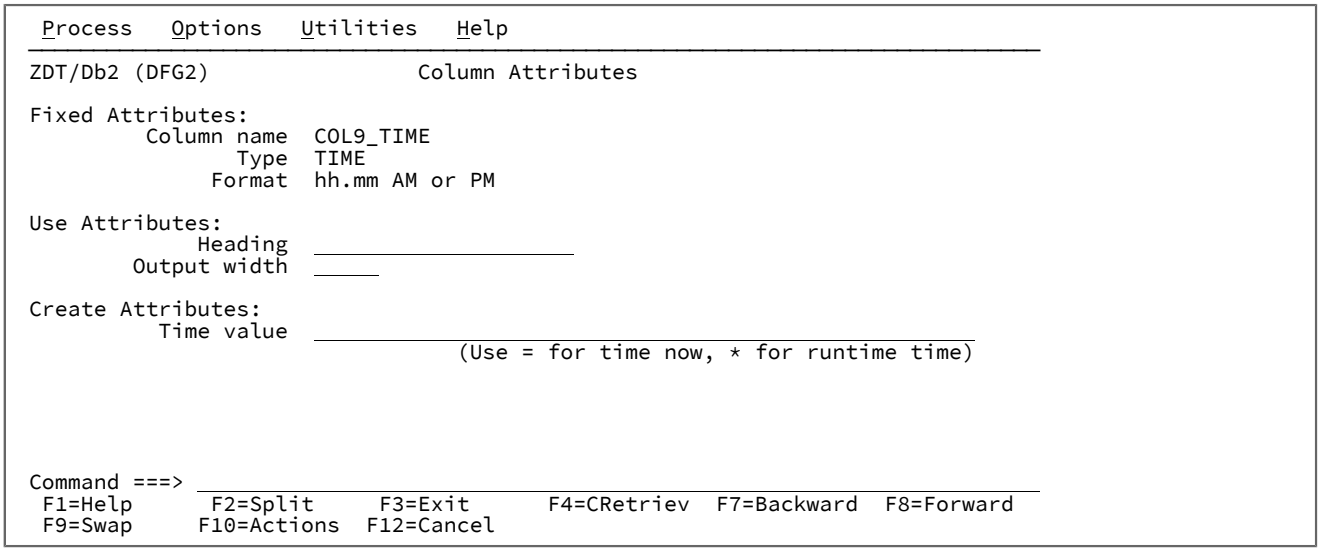
Related tasks

- Editing the template on page 245
- Changing column attributes on page 104
- Changing the attributes for a numeric column on page 107

Column Attributes panel (TIME)

The Column Attributes (TIME) panel describes the formatting and data create attributes for a TIME column.

Panel and field definitions



Column name

The name of the column.

Type

The Db2® data type of the column.

Format

The format in which the time is displayed or printed.

Heading

The alternative column heading.

Output width

The number of character positions used by view, edit, and print functions to show the column in TABL display or print format.

Time value

The initial value a column is to contain. Valid values are:

start time

A time in a valid Db2® or local format.

*** (asterisk)**

The time the data is created. Not valid if the Db2® installation time format is LOCAL.

=

The current time. Not valid if the Db2® installation time format is LOCAL.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

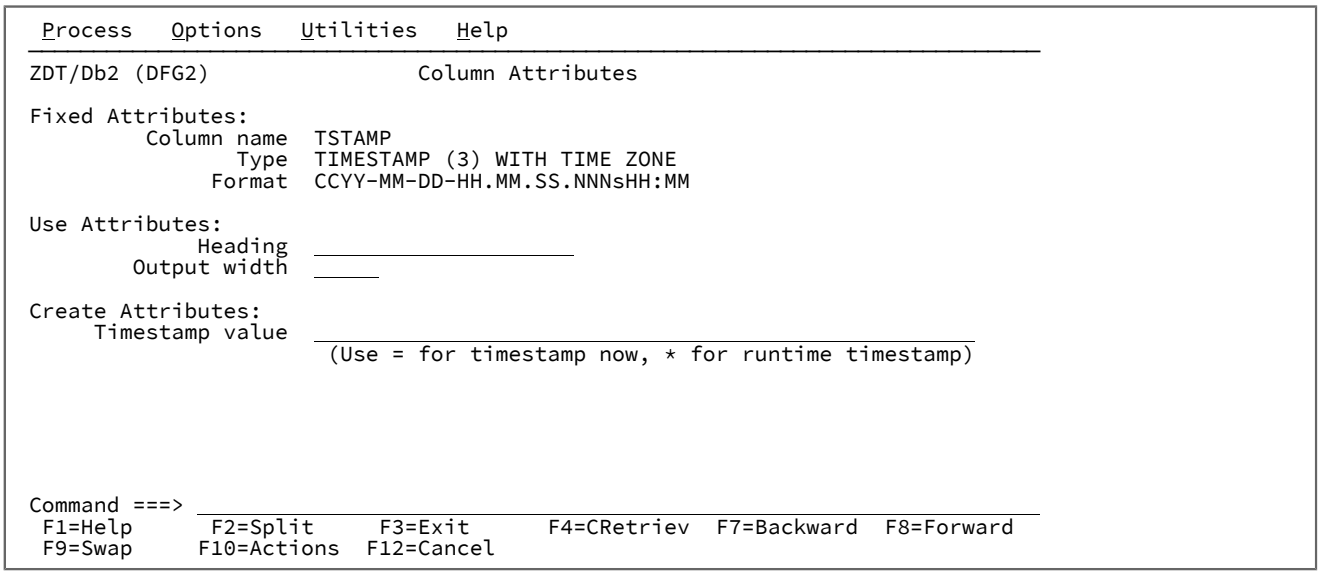
Related tasks

- Editing the template on page 245
- Changing column attributes on page 104
- Table 25: Date and Time field formats on page 443

Column Attributes panel (TIMESTAMP)

The Column Attributes (TIMESTAMP) panel describes the formatting and data create attributes for a TIMESTAMP column.

Panel and field definitions



The example shows a column defined as `TIMESTAMP(3) WITH TIME ZONE`. The Type value and the Format are different for columns with different TIMESTAMP data types.

Column name

The name of the column.

Type

The Db2® data type of the column.

Format

The format in which the timestamp is displayed or printed.

Heading

The alternative column heading.

Output width

The number of character positions used by view, edit, and print functions to show the column in TABL display or print format.

Timestamp value

The initial value a column is to contain. Valid values are:

start timestamp

A timestamp in the Db2® timestamp format.

*** (asterisk)**

The date and time the data is created.

=

The current date and time.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

Related tasks

- Editing the template on page 245
- Changing column attributes on page 104

Related references

- Db2 data types on page 440

Column Distribution panel

ZDT/Db2 displays the Column Distribution panel if you use the CDI line command against an item listed on the Tables, Views and Aliases panel on page 891.

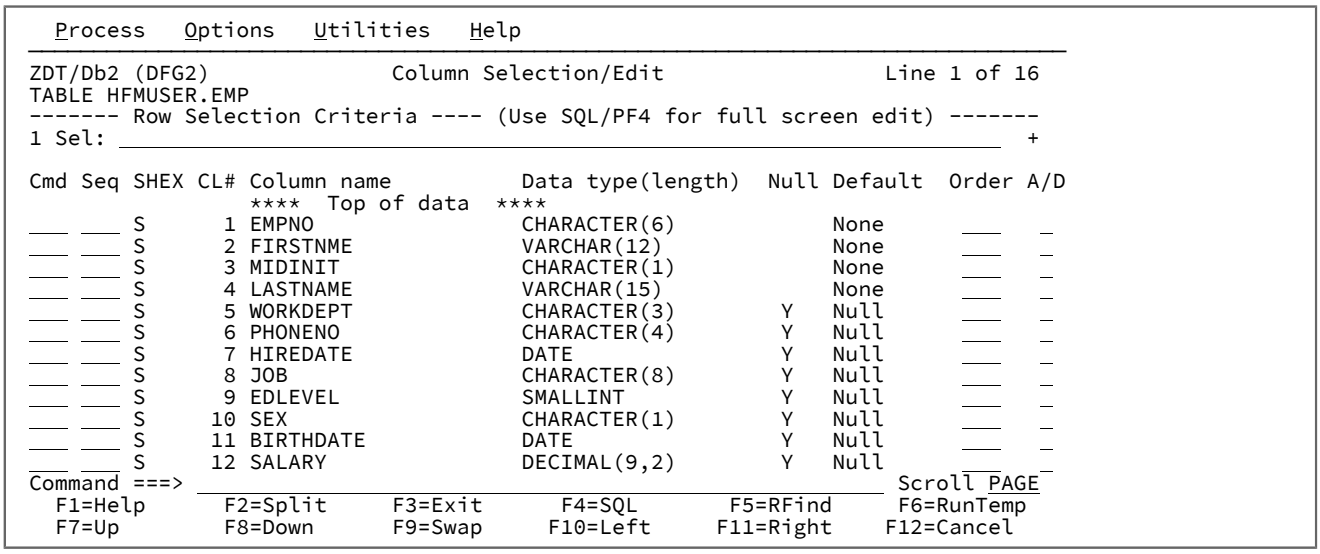
Column Part Statistics panel

ZDT/Db2 displays the Column Part Statistics panel if you use the PST line command against an item listed on the Columns panel on page 503.

Column Selection/Edit panel

You use the Column Selection/Edit panel to edit a template.

Panel and field definitions



Object type and name

The object type (TABLE or VIEW) and the name of the Db2® object being processed.

Row Selection Criteria

Field where you can enter a freeform SQL WHERE clause or expression that limits the rows selected from Db2®.

If the template already contains row selection criteria, ZDT/Db2 shows the corresponding WHERE clause in this field.

Cmd

Prefix command area for entering template editor prefix commands:

S, SS, S*n*, S*

Selecting (or deselecting) columns.

H, HH, H*n*

Holding (or freeing) columns.

E

Changing the attributes of a column.

X

Displaying data for character columns in LHEX. The data for Binary and Varbinary columns is displayed in LHEX by default.

Seq

The display sequence number of the column (in a Z Data Tools/Db2 editor session) indicating either of these:

- The sequence number you have entered to change the sequence of the displayed columns.
- The sequence number generated by ZDT/Db2 as a result of a hold template editor prefix command (H, HH, or H*n*) being entered in the **Cmd** prefix entry area.

S

Column selection indicator. An asterisk (*) shows that the column is selected. Only selected columns appear in a Z Data Tools/Db2 editor session.

H

Column hold indicator. An *"H"* shows that the column is held.

In TABL mode, held columns appear as the leftmost columns in a Z Data Tools/Db2 editor session and are unaffected by scrolling left or right. In SNGL mode, held columns appear as the topmost columns in an editor session and are unaffected by scrolling up or down.

E

Data edit indicator. An *"E"* shows that the column has been previously edited to add information such as data create, column headings, or scrambling options.

X

LHEX indicator. An *"X"* shows that the data for the column is displayed in LHEX (Long Hexadecimal) format.

Cl#

The column number as defined in the Db2® catalog. The column number reflects the Db2® column number unless either of the following are true:

- The column is a generated column
- The column is not a column of the first table named in the SELECT statement used to create the result table

If either of these conditions is true, the column has a number greater than the maximum Db2® column number, for all specified columns in the first Db2® object named in the SELECT statement.

Column name

The name of the column.

Data type(length)

The Db2® data type (and optionally the length) of the column.

Null

Shows whether the column is defined as nullable.

Y

Nullable column

(blank)

Db2® null values are not permitted

Default

Default value information for the column:

Always

Either a ROWID or identity column defined with the **generated always** attribute.

Default

Either a ROWID or identity column defined with the **generated by default** attribute.

None

There is no default value for the column.

Null

The default value is the Db2® null value.

On type

The column has a default value that depends on the data type of the column:

Data type

Default value

Numeric

0

Fixed-length character data types

Blanks

Varying-length data types

String of length 0

DATE

Current date

TIME

Current time

TIMESTAMP

Current date and time

On value

The column has a default value that is defined in the DEFAULTVALUE column of the SYSIBM.SYSCOLUMNS catalog table.

SQLID

The default value is the value of the SQL authorization ID of the process at the time the default value is used.

USER

The default value is the value of the USER special register at the time the default value is used.

Order

An optional sequence number indicating how to order the data, based on the data values for the column, when the data is retrieved from Db2®.

A/D

Sort sequence indicator. You can only specify this if an order sequence number (**Order**) is also specified.

A

Ascending

D

Descending

Parent panels

- Db2 View panel on page 631
- Db2 Edit panel on page 616
- Print Utility panel on page 764
- Copy Utility ("From") panel on page 513
- Export Utility ("From") panel on page 698
- Data Create Utility panel on page 603

Child panels

To display this panel…	Use/do this
Table View panel on page 882	If parent panel is Db2® View and you press F3
Db2 View panel on page 631	If parent panel is Db2® View and you press F12
Table Edit panel on page 868	If parent panel is Db2® Edit and you press F3
Db2 Edit panel on page 616	If parent panel is Db2® Edit and you press F12
Table Browse panel on page 866	If parent panel is Db2® Browse and you press F3

To display this panel…	Use/do this
Db2 Browse panel on page 612	If parent panel is Db2® Browse and you press F12
Template Save panel on page 902	Press F3
Db2 Template Description panel on page 626	Primary command DESCRIBE
Export Utility ("To") panel on page 701	If parent panel is Export Utility and you press F3
Column Attributes panel (alphanumeric) on page 484	Line command E for an alphanumeric column
Column Attributes panel (numeric) on page 490	Line command E for a numeric column
Column Attributes panel (DATE) on page 489	Line command E for a DATE column
Column Attributes panel (TIME) on page 494	Line command E for a TIME column
Column Attributes panel (TIMESTAMP) on page 495	Line command E for a TIMESTAMP column
Row Selection Criteria panel on page 799	Enter `1` on the command line

Related tasks

- Locating a column on page 142
- Restricting the data that ZDT/Db2 loads from Db2 on page 138
- Large numeric fields on page 168
- Copying data from one Db2 object to another on page 246
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279
- Editing the template on page 245
- Running Print in batch on page 331
- Editing a template on page 84
- Using the SQL Edit session on page 95
- Manipulating columns on page 98
- Selecting columns on page 99
- Holding columns on the display when scrolling on page 100
- Changing the sequence of displayed or printed columns on page 101
- Changing the sequence of displayed or printed rows on page 103

- Changing the column heading on page 105
- Changing column attributes on page 104

Related references

- Db2 data types on page 440
- TEDIT primary command on page 1006

Columns panel

You use the Columns panel to list column object types in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSCOLUMNS.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                      Columns
                                                          Format TABL

SEL    NAME                TBCREATOR TBNAME               COLNO COLTYPE  LENGTH
       *                   *         *                        * *        *
----   #1--+----1----+---  #3--+----  #2--+----1----+---  ----#4 #5--+---  ----#6
****  Top of data  ****
_____  MAJSYS              CARLAND   PMR91932                1 CHAR          1
_____  ACTION              CARLAND   PMR91932                2 CHAR          1
_____  OBJFLD              CARLAND   PMR91932                3 CHAR          2
_____  SRCHCRIT            CARLAND   PMR91932                4 CHAR          2
_____  SCRTYPE             CARLAND   PMR91932                5 CHAR          1
_____  HEADTXT             CARLAND   PMR91932                6 CHAR         50
_____  SELTXT              CARLAND   PMR91932                7 CHAR         50
_____  INFOTXT             CARLAND   PMR91932                8 CHAR         79
_____  HELPTXT             CARLAND   PMR91932                9 CHAR         79
_____  PFKTXT1             CARLAND   PMR91932               10 CHAR         79
_____  PFKTXT2             CARLAND   PMR91932               11 CHAR         79
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Constraint Columns for panel on page 509
- Columns in Table panel on page 505
- Privileges panels on page 768

Child panels

To display this panel…	Use/do this
Tables, Views and Aliases panel on page 891	Line command AT
See Comment panels on page 506	Line command COM
Column Distribution panel on page 497	Line command DI
Distinct Types panel on page 637	Line command DT
Grant privileges panels on page 715	Line command G
Columns panel on page 503	Line command H
Details panels on page 636	Line command I
	Line command LAB

To display this panel…	Use/do this
Label Column panel on page 728	
	Line command P
Privileges panels on page 768	
	Line command ROW
Display Row panel on page 637	
	Line command T
Tables, Views and Aliases panel on page 891	
	Line command X
Indexes panel on page 723	
	Line command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Columns in Index panel

ZDT/Db2 displays the Columns in Index panel if you use the COL line command against an item listed on the Indexes panel on page 723.

Columns in Table panel

ZDT/Db2 displays the Columns in Table panel if you use the COL line command against an item listed on:

- The Tables, Views and Aliases panel on page 891, or
- The Table Check Constraint Dependencies panel on page 868

Comment panels

These panels are:

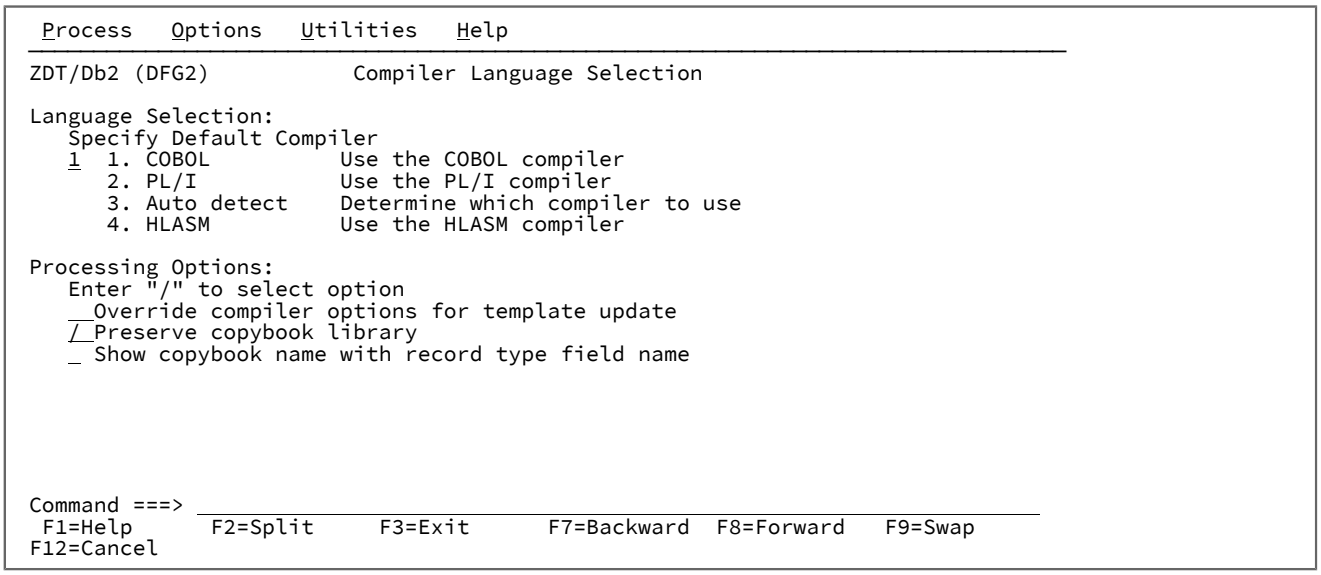
- Comment Column panel
- Comment Distinct Type panel
- Comment Procedure panel
- Comment Specific Function panel
- Comment Table panel
- Comment Trigger panel

ZDT/Db2 displays one of these panels when you specify the COM line command against an object displayed in a list of Db2® objects.

Compiler Language Selection panel

You use the Compiler Language Selection panel to set the language to be used for compiling the current source member to create a template. The default language is COBOL. This option is saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



Language Selection

The following language can be specified for the compilation of the source member to create a template:

1. COBOL

Use the COBOL compiler. This is the default setting.

2. PL/I

Use the PL/I compiler.

3. Auto detect

Perform a heuristic analysis of the source and use the appropriate compiler.

4. HLASM

Use the HLASM compiler.

Processing Options

Override compiler options for template update

When selected, overrides the compiler options found in the template being updated with the current compiler options.

Foreground processing always overrides the compiler options for older templates that do not contain compiler options.

You should select this option for batch template update of older templates to generate compiler options in the JCL.

Preserve copybook library

Ensures that, if a copybook still exists in the library that it was previously found in and that library is in the list that the update is using, then that version of the copybook is used.

If you do not select this option, or the copybook no longer exists in the library it was previously found in, or that library is not in the list the update is using, then the utility searches the libraries in the order they are listed and uses the first version of the copybook that it finds.

Show copybook name with record type field name

Select this option to make the copybook name for a record layout visible during template edit and for an edit, view or browse session where a copybook or template is being used.

Note: For the editor session where the layout name is specified it is prefixed with the name of the associated copybook. This may cause the layout name to be truncated, depending on its length.

Parent panels

- Set System Processing Options panel on page 835

Child panels

None.

Related tasks

- Compiler language selection (option 0.4) on page 65
- How ZDT/Db2 compiles a copybook into a template on page 80

Constraint Columns for panel

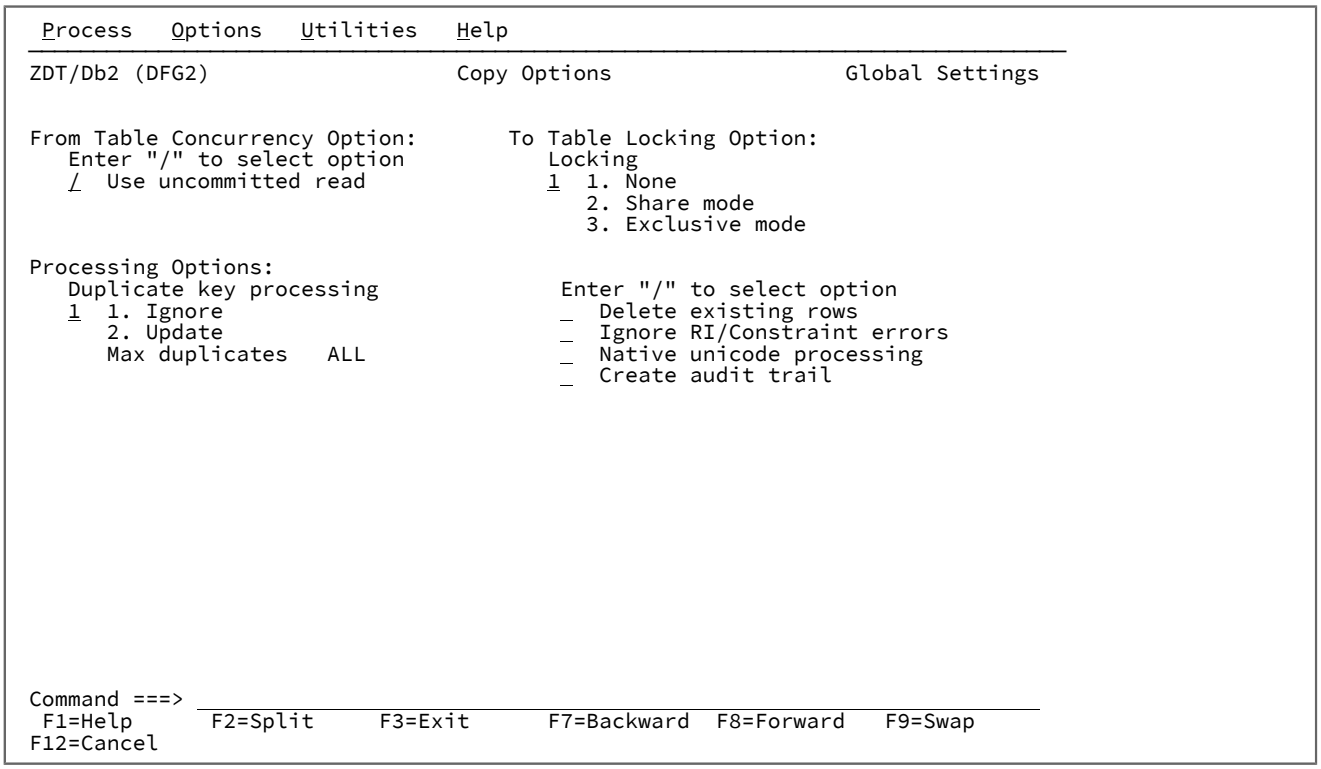
ZDT/Db2 displays the Constraint Columns for panel if you use the COL line command against an item listed on the Table Check Constraints panel on page 868.

Copy Options panel

You use the Copy Options panel to set the global (or initial) options used by the Copy utility (3.3). These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

The global Copy options are restored at the start of each Copy session and can be temporarily changed for a copy session by selecting **Edit Copy Options** on the Copy *"To"* entry panel.

Panel and field definitions



Use uncommitted read

Indicates whether to enable retrieval of uncommitted data for the Copy session:

/

Retrieve data, including data not yet committed by another user or application program. The resultant SELECT statement used by ZDT/Db2 to retrieve table data has the suffix *"WITH UR"* appended. This allows reading through any locks which might be held on that table. The default is not selected.

(blank)

Only retrieve committed data from tables without write locks.

Note: If ZDT/Db2 attempts to retrieve data that has a write lock, an SQL error can result.

Locking

ZDT/Db2 provides options to lock the table to prevent other users from either updating or reading data. Use these options with caution.

1. **None**

   ZDT/Db2 does not issue an explicit table lock for the object being copied. Locking of rows and pages within the object is managed by Db2® according to ZDT/Db2 concurrency options and Db2® installation and object definition options. This is the default setting.

2. **Share mode**

   ZDT/Db2 issues a LOCK TABLE IN SHARE MODE statement before reading the data from Db2®. This restricts all other users to read-only operations on the object.

3. **Exclusive mode**

   ZDT/Db2 issues a LOCK TABLE IN EXCLUSIVE MODE statement before reading the data from Db2®. This prevents all other users from accessing the object, unless the other process is running with an isolation level of uncommitted read.

Duplicate key processing

You can specify what action ZDT/Db2 takes if it encounters a duplicate key. Set this option to one of the following:

1. **Ignore**

   Existence of duplicate key ignored. This is the default.

2. **Update**

   Updates duplicate rows when Db2® returns a duplicate key error (SQL error code -803) in response to an INSERT SQL statement. When this option is selected, ZDT/Db2 attempts to update the existing row. All columns of the table that are part of at least one unique index are used to identify the row. No comparison is made between the copied row and the existing row prior to the update operation (that is, the update is done regardless of whether the copied row and existing row are the same). When using this option, it is recommended that a Db2® table name, rather than a view name, be specified. ZDT/Db2 does not attempt to update existing rows in a view defined on a view.

For _____ duplicates

Specifies the maximum number of duplicates permitted. Valid values are:

- A number in the range 0 to 2147483647. If ZDT/Db2 detects more than this number of duplicates during the copy process, the copy is canceled and all changes are rolled back.
- ALL. ALL is the default.

Delete existing rows

Determines whether ZDT/Db2 deletes all existing rows in the table before copying the data.

/

Delete all existing rows.

(blank)

Do not delete existing rows. This is the default setting.

Ignore RI/Constraint errors

Determines what processing occurs when the Copy utility encounters an SQLCODE-530 (RI error - no primary key) or SQLCODE-545 (Constraint error).

/

The Copy process ignores the error and continues with the next row. The row is not copied or updated and is not included in the copy count.

(blank)

Copy utility canceled. This is the default setting.

Native unicode processing

Determines how ZDT/Db2 processes Unicode encoded data when copying data between two Unicode encoded tables. When the option is enabled, Unicode encoded data is copied between Unicode encoded tables without the potential for character substitution. When the option is not enabled, the option has no effect; that is, the default processing is used. When either or both of the FROM or TO tables are not Unicode encoded, the option has no effect, even if enabled. When this option and the batch execution options are enabled, "NATIVE= YES" is shown in the generated JCL.

/

Process Unicode data natively.

(blank)

Do not process Unicode data natively. This is the default setting.

Create audit trail

Determines whether audit records are written for the current copy operation. This option may not appear depending on the installation choices made by the product installer. When the option is available, the default is to not write audit records.

Parent panels

- Setting options for the current Copy session on page 251
- Copy Utility ("To") panel on page 521

Child panels

None.

Related tasks

- Setting options for the current Copy session on page 251
- Copy utility options (option 3.3) on page 68

Copy Utility ("From") panel

You use the Copy Utility (“From”) panel to enter the name of an object to be copied.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Copy Utility

From Db2 Object:
   Location  . . . . . _______________        Database . . ________ (optional)
   Owner . . . . . . . DSN8810_______  +      Table space  ________ (optional)
   Name  . . . . . . . EMP_____________________________________________  +

   Copy count  . . . . ALL_____       Number of rows to copy

From Template:
   Data set name . . . _____________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                   Enter "/", "A"lways to select option
   3  1. Above                      _  Edit template
      2. Previous                   _  Copy panel values
      3. Generate from table        _  Create audit trail (*)
      4. Generate/Replace
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Copy count

The maximum number of rows to be copied from a Db2® object. It must be one of the following:

- A number in the range 1–99999999,
- An asterisk (*), or
- ALL. This is the default, where all rows selected from the Db2® object are copied.

When the *"From"* template includes a *"Row Selection Criteria"* (or WHERE clause), only those rows matching the selection criteria are considered eligible to be copied.

Data set name

The name of a data set that contains, or will contain, the template. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The Copy Utility panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. **Generate/Replace**

   Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **From Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following option by entering a / or an "A":

Edit template

Use to edit the template before copying the data. The default is to not edit the template.

Copy panel values

Use to copy the Db2® object name and template details to the *"To"* panel. The default is to not copy any values to the "To" panel.

Create audit trail

Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Parent panels

- Utility Functions panel on page 915
- Tables, Views and Aliases panel on page 891

Child panels

To display this panel…	Use/do this
Column Selection/Edit panel on page 497	Select **Edit template**
Copy Utility ("To") panel on page 521	Press Enter

Related concepts

- Information stored in a template on page 75
- SAF-rule controlled auditing on page 71

Related tasks

- Selecting options on ZDT/Db2 panels on page 45
- Working with templates on page 74
- Where you can use templates on page 22
- Copying data from one Db2 object to another on page 246
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279
- Specifying a data set and a member name on page 52
- ZDT/Db2 system options (option 0.2) on page 65

COPY Utility (Index Spaces) panel

You use the COPY Utility (Index Spaces) panel to generate utility control statements using the COPY INDEXSPACE form of the COPY utility.

The format of the COPY Utility (Index Spaces) panel depends on which, if any, of the filter fields you specified on the Db2® Utilities panel:

- The panel shown in Figure 118: COPY Utility (Index Spaces) panel—default format on page 518 is displayed by default, *or* if you specified information in the **Database name** filter field, the **Table/index space name** filter field, or both.
- The panel shown in Figure 119: COPY Utility (Index Spaces) panel—alternative format on page 518 is displayed if you specified information in the **Table/index owner** filter field, the **Table/index name** filter field, or both.

Note: Indexes need to have been defined with the COPY YES clause. Indexes which have not been defined with the COPY YES clause cannot be selected for image copying in the panel.

Panel and field definitions

Figure 118. COPY Utility (Index Spaces) panel—default format

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           COPY Utility (Index Spaces)           Row 1 to 8 of 8


S P          Index
e R Database Space    Part'n       Allocation
l C Name     Name     Number     Primary Secdry       Space
_   BILLSDB1 XDEPT1        0           3      3           0
_   BILLSDB1 XDEPT2        0           3      3           0
_   BILLSDB1 XDEPT3        0           3      3           0
_   BILLSDB1 XEMP1         1           3      3           0
_   BILLSDB1 XEMP1         2           3      3           0
_   BILLSDB1 XEMP1         3           3      3           0
_   BILLSDB1 XEMP1         4           3      3           0
_   BILLSDB1 XEMP2         0           3      3           0
****************************** Bottom of data ********************************


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 119. COPY Utility (Index Spaces) panel—alternative format

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           COPY Utility (Index Spaces)           Row 1 to 8 of 8


S P
e R Index    Index             Part'n       Allocation
l C Creator  Name              Number     Primary Secdry       Space
_   ARROWSB  XDEPT1                 0           3      3           0
_   ARROWSB  XDEPT2                 0           3      3           0
_   ARROWSB  XDEPT3                 0           3      3           0
_   ARROWSB  XEMP1                  1           3      3           0
_   ARROWSB  XEMP1                  2           3      3           0
_   ARROWSB  XEMP1                  3           3      3           0
_   ARROWSB  XEMP1                  4           3      3           0
_   ARROWSB  XEMP2                  0           3      3           0
****************************** Bottom of data ********************************


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Sel

Selection column. To select an index space, type S in this column. Selected index spaces show an asterisk (*) in the **PRC** column.

PRC

Processing indicator. An asterisk (*) indicates the index space is selected.

Database Name

The name of the database in which the index space is defined.

Index Space Name

The name of the index space.

Part'n Number

The partition number for the table space. This is 0 for a non-partitioned table space. A non-zero value indicates a partition of a partitioned table space. By default, when you select a non-partitioned table space for processing, the utility control statements do not include the DSNUM ALL parameter. When you select a partition, the utility control statements include a DSNUM *nnn* clause (*nnn* is the partition number you selected).

Primary

The primary allocation value for the table space. See the *DB2 Utility Guide and Reference* for a full explanation. This value gives some indication of the size of the table space (but is not definitive).

Secondary

The secondary allocation value for the table space. See the *DB2 Utility Guide and Reference* for a full explanation.

Space

The space allocation for the table space, as reported by the STOSPACE utility. See the *DB2 Utility Guide and Reference* for a full explanation. This value gives an accurate indication of the size of the table space when the STOSPACE utility was run.

Index Creator

The name of the index creator.

Index Name

The name of the index.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

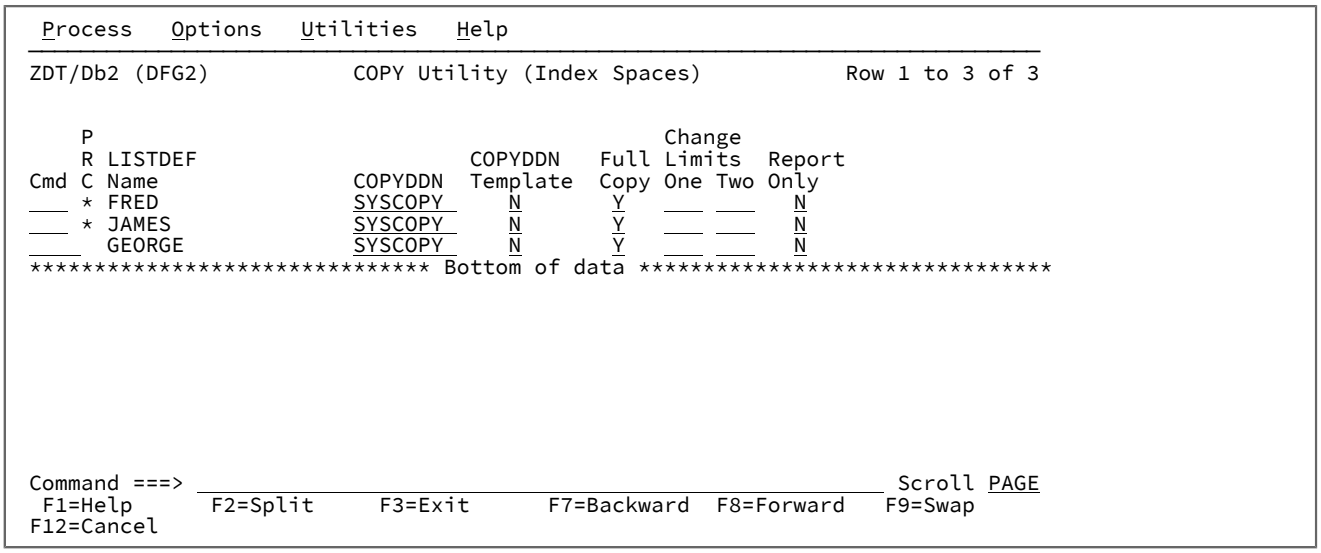
Related tasks

- COPY utility on page 354
- COPY (index spaces) on page 356

COPY Utility (Index Spaces) with LISTDEF panel

You use the COPY Utility (Index Spaces) with LISTDEF panel to generates utility control statements using the COPY TABLESPACE form of the COPY utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- COPY utility on page 354
- COPY (index spaces) on page 356

Copy Utility ("To") panel

You use the Copy Utility (“To”) panel to enter the name of the Db2® object that is the target of the copy operation. You also use this panel to specify the name of a template that is to provide the column mapping and initialization information.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
Copy from DSN8810.EMP

To Db2 Object:
   Location  . . . . . _________________          Database . . _________ (optional)
   Owner . . . . . . . HFMUSER          +         Table space  _________ (optional)
   Name  . . . . . . . EEMP                                                 +


To Template:
   Data set name . . . _____________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                    Enter "/", "A"lways to select option
   3  1. Above                       _  Edit copy options
      2. Previous                    _  Edit template mapping
      3. Generate from table         _  Batch execution
      4. Generate/Replace
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Data set name

The name of a data set that contains, or will contain, the template. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The Copy Utility panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. Generate from table

Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **To Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a / or an "A":

Edit copy options

Displays the copy options panel. The default is not selected.

Edit template mapping

Use to edit the template for the table to which you are copying the data, before copying the data. The default is not selected.

Batch execution

Generates the JCL to perform the ZDT/Db2 Copy function as a batch job. You can edit the JCL and submit the job manually once the JCL has been generated. The default is not selected - the Copy job runs in the foreground.

Parent panels

- Copy Utility ("From") panel on page 513

Child panels

To display this panel…	Use/do this
Copy Options panel on page 509	Select **Edit copy options**
Template Mapping panel on page 900	Select **Edit template mapping**

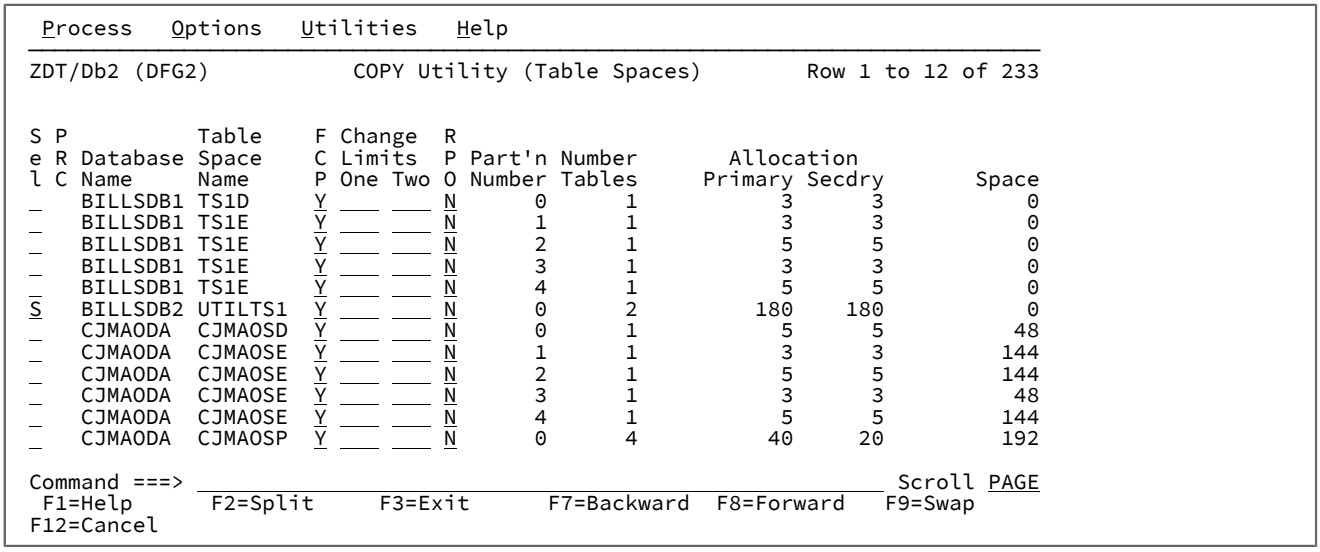
Related tasks

- Specifying a data set and a member name on page 52
- Selecting options on ZDT/Db2 panels on page 45
- Copying data from one Db2 object to another on page 246
- Mapping data on page 112
- Copying data from one Db2 object to another on page 246

COPY Utility (Table Spaces) panel

You use the COPY Utility (Table Spaces) panel to generate utility control statements using the COPY TABLE SPACE form of the COPY utility.

Panel and field definitions



```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           COPY Utility (Table Spaces)        Row 1 to 12 of 233


S P          Table    F Change  R
e R Database Space    C Limits  P Part'n Number       Allocation
l C Name     Name     P One Two O Number Tables     Primary Secdry        Space
_   BILLSDB1 TS1D     Y ___ ___ N      0      1           3      3            0
_   BILLSDB1 TS1E     Y ___ ___ N      1      1           3      3            0
_   BILLSDB1 TS1E     Y ___ ___ N      2      1           5      5            0
_   BILLSDB1 TS1E     Y ___ ___ N      3      1           3      3            0
_   BILLSDB1 TS1E     Y ___ ___ N      4      1           5      5            0
S   BILLSDB2 UTILTS1  Y ___ ___ N      0      2         180    180            0
_   CJMAODA  CJMAOSD  Y ___ ___ N      0      1           5      5           48
_   CJMAODA  CJMAOSE  Y ___ ___ N      1      1           3      3          144
_   CJMAODA  CJMAOSE  Y ___ ___ N      2      1           5      5          144
_   CJMAODA  CJMAOSE  Y ___ ___ N      3      1           3      3           48
_   CJMAODA  CJMAOSE  Y ___ ___ N      4      1           5      5          144
_   CJMAODA  CJMAOSP  Y ___ ___ N      0      4          40     20          192

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Sel

Selection column. To select a table space, type S in this column. Selected table spaces show an asterisk (*) in the **PRC** column.

PRC

Processing indicator. An asterisk (*) indicates the table space is selected.

Database Name

The name of the database in which the table space is defined.

Table Space Name

The name of the table space.

FCP

Full Copy flag:

Y

Take a full image copy. This is the default setting.

N

Take a partial image copy.

Change Limits (One Two)

You can specify 1 or 2 values indicating the percentage of changed pages needed to trigger a partial image copy, and a full image copy. See the *DB2 Utility Guide and Reference* for a detailed explanation. A value entered in either of the **Change Limits** fields overrides the **FCP** setting.

RPO

Report Only. This only takes effect if you have specified a change limit value (see **Change Limits**).

Y

Only image copy information is displayed. Image copies are not taken, only recommended. ZDT/Db2 generates a DD DUMMY in the JCL for the SYSCOPY DD name.

N

Image copy information is displayed and image copies taken. This is the default setting.

The remaining columns of the ISPF table are display-only fields showing information from the Db2® catalog.

Part'n Number

The partition number for the table space. This is 0 for a non-partitioned table space. A non-zero value indicates a partition of a partitioned table space. By default, when you select a non-partitioned table space for processing, the utility control statements do not include the DSNUM ALL parameter. When you select a partition, the utility control statements include a DSNUM *nnn* clause (*nnn* is the partition number you selected).

Number Tables

The number of tables defined within the table space.

Primary

The primary allocation value for the table space. See the *DB2 Utility Guide and Reference* for a full explanation. This value gives some indication of the size of the table space (but is not definitive).

Secdry

The secondary allocation value for the table space. See the *DB2 Utility Guide and Reference* for a full explanation.

Space

The space allocation for the table space, as reported by the STOSPACE utility. See the *DB2 Utility Guide and Reference* for a full explanation. This value gives an accurate indication of the size of the table space when the STOSPACE utility was run.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- COPY utility on page 354

COPY Utility (Table Spaces) with LISTDEF panel

You use the COPY Utility (Table Spaces) with LISTDEF panel to generate utility control statements using the COPY TABLE SPACE form of the COPY utility, and with a LISTDEF provided.

Panel and field definitions

```
 Process   Options   Utilities   Help
-------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           COPY Utility (Table Spaces)           Row 1 to 1 of 1


    P                                          Change
    R LISTDEF                    COPYDDN   Full Limits  Report
Cmd C Name                COPYDDN  Template  Copy One Two Only
      FRED                SYSCOPY     N        Y   ___ ___   N
      JAMES               SYSCOPY     N        Y   ___ ___   N
****************************** Bottom of data ********************************









Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

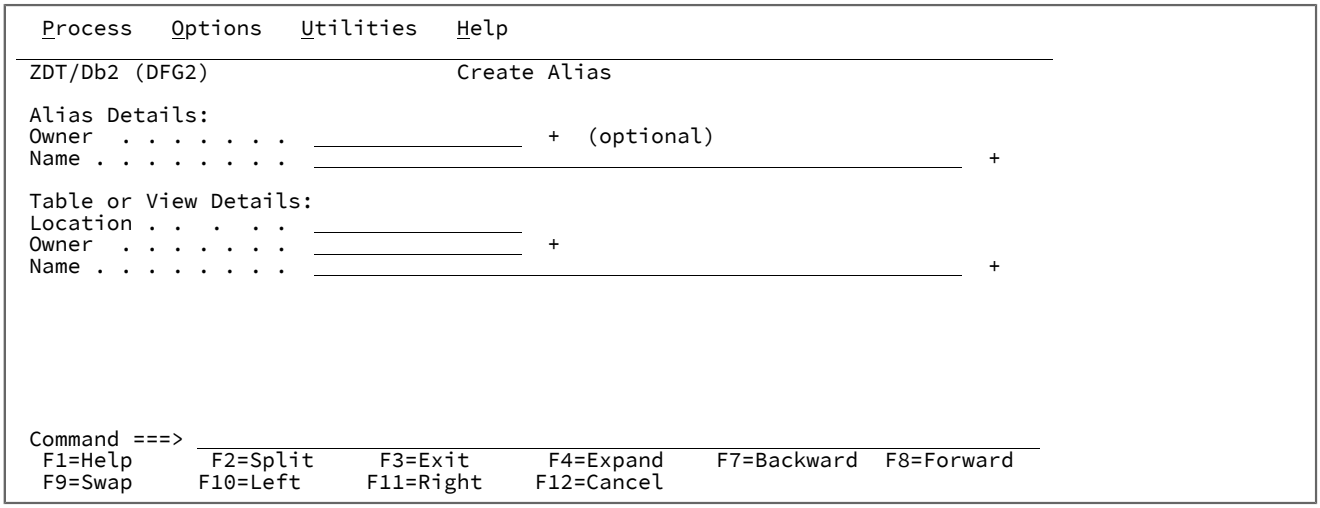
Related tasks

- COPY utility on page 354
- COPY (table spaces) on page 355

Create Alias panel

You use the Create Alias panel to create an alias for a table or a view at the correct location.

Panel and field definitions



Owner (Alias Details)

The owner of the alias you want to create (optional). If you leave this field blank, ZDT/Db2 uses the current SQL ID as a default ID. Otherwise, type the full name of the owner.

Name Alias Details

The name of the alias you want to create (mandatory).

Location

The location of the table or view for which you want to create an alias (optional). If you leave this field blank, ZDT/Db2 uses the location of the local Db2® system.

Owner (Table or View Details)

The owner of the table or view for which you want to create an alias (optional). If you leave this field blank, ZDT/Db2 uses the current SQL ID as a default ID. Otherwise, type the full name of the owner.

Name (Table or View Details)

The name of the table or view for which you want to create an alias (mandatory).

Parent panels

- Db2 Object Functions panel on page 622
- Tables, Views and Aliases panel on page 891
- Synonyms panel on page 855

Child panels

None.

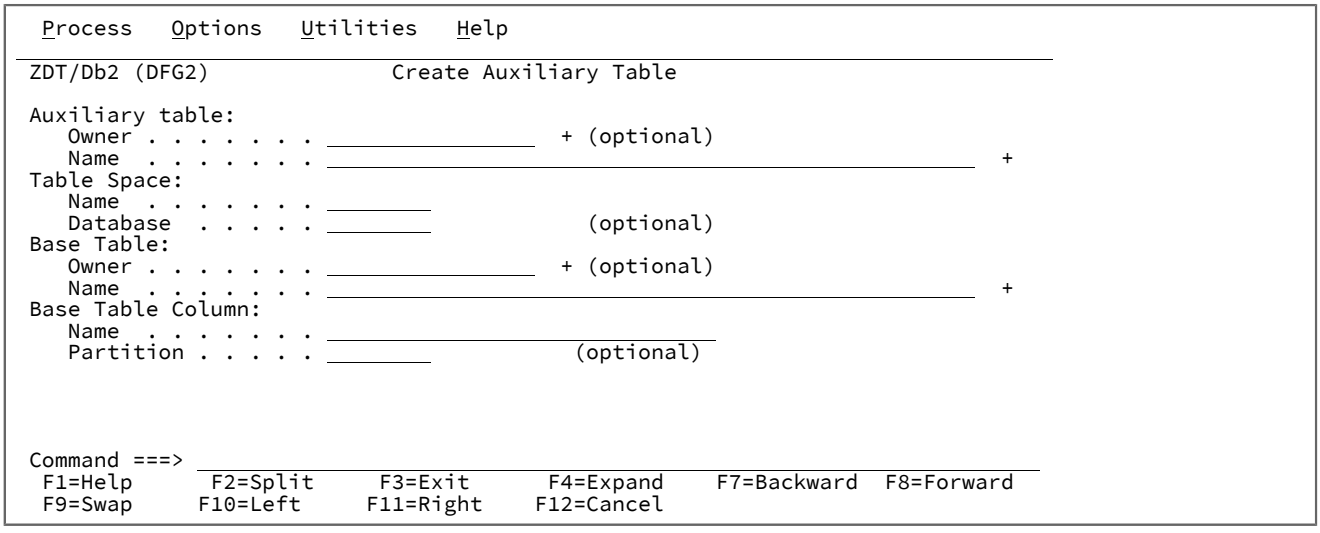
Related tasks

- Creating and dropping Db2 objects on page 208
- Creating an alias on page 236

Create Auxiliary Table panel

You use the Create Auxiliary Table panel to create a new auxiliary table.

Panel and field definitions



Owner (Auxiliary Table)

Specifies the owner of the auxiliary table.

Name (Auxiliary Table)

Specifies the name of the auxiliary table.

Name (Table Space)

Specifies the LOB table space in which the auxiliary table is created.

Database (Table Space)

Specifies the database to which the specified table space belongs.

Owner (Base Table)

Specifies the owner of the base table containing the LOB column that is stored in the auxiliary table.

Name (Base Table)

Specifies the name of the base table containing the LOB column that is stored in the auxiliary table.

Name (Base Table Column)

Specifies the name of the LOB column in the base table that is stored in the auxiliary table.

Partition (Base Table Column)

For base tables defined in partitioned table spaces, specifies the partition of the base table for which this auxiliary table stores data.

Parent panels

- Db2 Object Functions panel on page 622

Child panels

None.

Related tasks

- Creating and dropping Db2 objects on page 208
- Creating an auxiliary table on page 243

Create Database panel

You use the Create Database panel to create a new database.

Panel and field definitions

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                 Create Database

Database:
   Name . . . . . . . . . . . _________    (required)

Buffer Pools, Storage Groups:
   Storage group name . . . . . _________  + (optional)
   Buffer pool for table spaces _________    (optional)
   Buffer pool for indexes      _________    (optional)

Database Options:
   Database Usage                      Data Encoding (only for user databases)
   _  1. User database   (default)     _  1. EBCDIC   (default)
      2. Work file database              2. ASCII
      3. Temporary database              3. UNICODE

Member Selection:
   Member name                 _________   (only valid for work file databases)

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Name

Specify the name of the Db2® database you want to create.

Storage group name

Names the Db2® storage group to be used, as the default group for DASD allocation for table spaces and indexes in the database.

Buffer pool for table spaces

Specifies the default buffer pool name to be used for table spaces created within the database.

1. 8K buffers: (BP8K0-BP8K9)
2. 16K buffers:(BP16K0-BP16K9)

Note:

8K and 16K buffer pools are not supported for work file databases.

Buffer pool for indexes

Specifies the default buffer pool name to be used for the indexes created within the database.

Database Usage

1. User database (default)

Specifies the database is an ordinary database for general user tables and indexes.

2. Work file database

Specifies the database is a work file database for use in a data sharing environment.

Temporary database

Specifies the database is for declared temporary tables only.

Data Encoding (only for user databases)

1. EBCDIC (default)

Specifies the encoding scheme.

2. ASCII

Specifies the encoding scheme.

3.UNICODE

Specifies the encoding scheme.

Parent panels

- Db2 Object Functions panel on page 622
- Databases panel on page 609

Child panels

None.

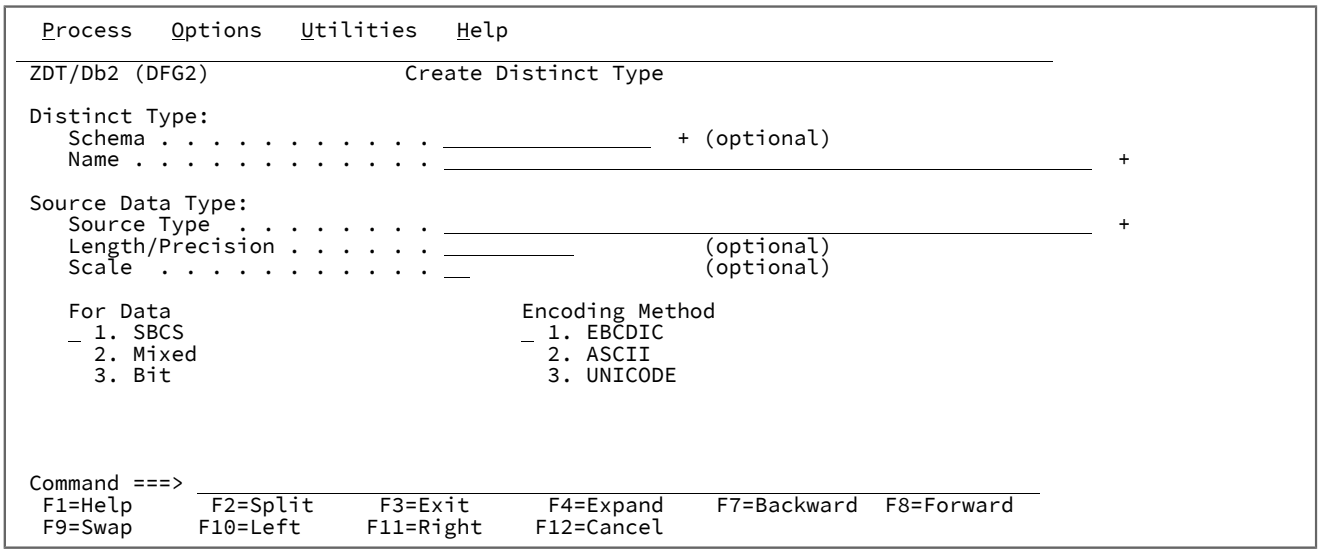
Related tasks

- Creating and dropping Db2 objects on page 208
- Creating a database on page 223

Create Distinct Type panel

You use the Create Distinct Type panel to create a new distinct type.

Panel and field definitions



Schema

The schema name in which to create the function.

Name

The name of the user-defined function.

Source Type

Db2® data type in which the distinct type is based.

Length/Precision

Data length of the distinct type. For distinct types based on the DECIMAL data type, this is the precision.

Scale

For distinct types based on the DECIMAL data type.

For Data

For non-DBCS data types (CHAR, VARCHAR and CLOB), specifies the subtype. BIT is not allowed for CLOB.

Encoding Method

1. EBCDIC (default)

Specifies the encoding scheme for the distinct data type.

2. ASCII

Specifies the encoding scheme for the distinct data type.

3.UNICODE

Specifies the encoding scheme for the distinct data type.

EBCDIC and ASCII specifies the encoding scheme for the distinct data type.

Parent panels

- Db2 Object Functions panel on page 622
- Distinct Types panel on page 637

Child panels

None.

Related tasks

- Creating and dropping Db2 objects on page 208
- Creating a distinct type on page 239

Create Function: Option List (1/2) panel

You use the Create Function: Option List (1/2) panel to specify information to define options for the function to be created.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)        Create Function: Option List (1/2)

   External Name . . . _____________________________  + (usually optional)
   Scratchpad  . . . . _____                           (optional)
   Package Collection  _____________________________  + (optional)
   WLM Environment . . _____________________________  + (optional)
   Time Limit  . . . . __________                      (CPU units - optional)

   Language (Required)    SQL                      Security
   _  1. ASSEMBLE         _  1. Reads SQL Data     _  1. Db2
      2. C                   2. No SQL                2. User
      3. COBOL               3. Modifies SQL Data     3. Definer
      4. PLI                 4. Contains SQL

   Parallelism
   _  1. Allowed
      2. Disallowed

   Run-time Options (optional)
   ____________________________________________________________________________ +
Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

External Name

Name of the MVS™ load module that contains the user-written code to implement the function.

Scratch Pad

Size of the scratch pad allocated by Db2® and passed to the function when it is called.

Package Collection

Specifies the package collection to be used when the function is run. If not specified, the invoking program's package collection is used.

Workload Manager Environment

Specifies the MVS™ workload manager environment in which the function is to run.

Time Limit (CPU units)

Limit on processor service units used by a single invocation of the function.

LANGUAGE (Required)

Application programming language of the function:

SQL

Specifies what type of SQL statements the function contains:

Security

Specifies the interaction of the function with an external security product:

Parallel

Specifies if Db2® can consider parallelism for the function:

Run-time options

Specifies the Language Environment® run-time options to be used for the function.

Parent panels

- Create Function panel on page 536

Child panels

None.

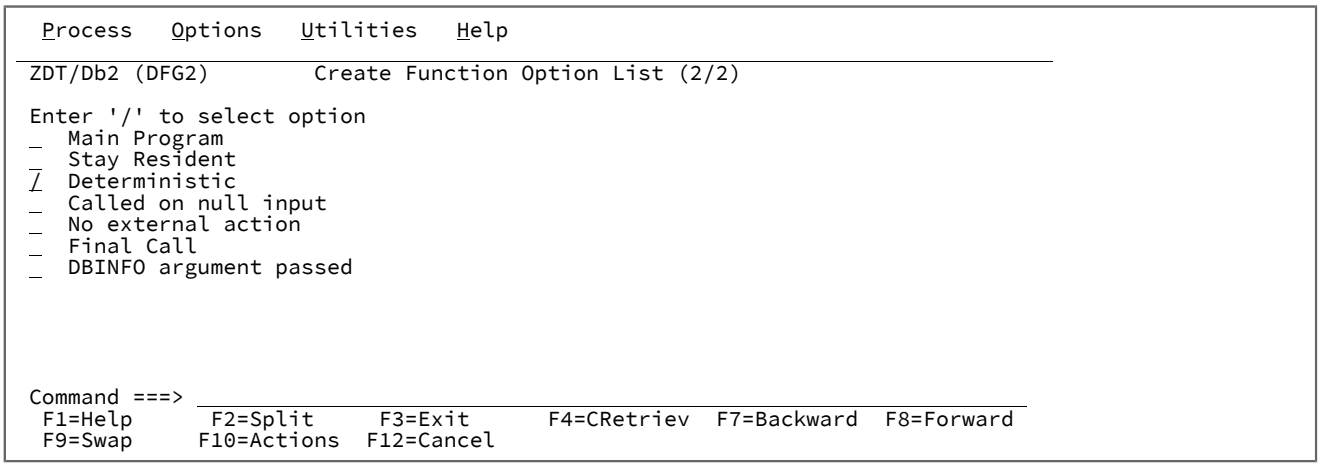
Related tasks

- Creating a function on page 239

Create Function: Option List (2/2) panel

You use the Create Function: Option List (2/2) panel to further specify information to define options for the function to be created.

Panel and field definitions



You can select the following options by entering a "/":

Main Program

The program runs as a main routine, otherwise it runs as a subroutine.

Stay Resident

The function load module remains in memory when the function ends.

Deterministic

The function always returns the same result for the same set of input values.

Called on Null Input

The function is called even if one or more of the input parameters has a null value.

No external action

The function does not take any action that changes the state of an object Db2® does not manage.

Final Call

A final call is made to the function to allow it to perform any required cleanup operations.

DBINFO Argument Passed

Additional Db2® information is passed when the function is called.

Parent panels

- Create Function panel on page 536

Child panels

None.

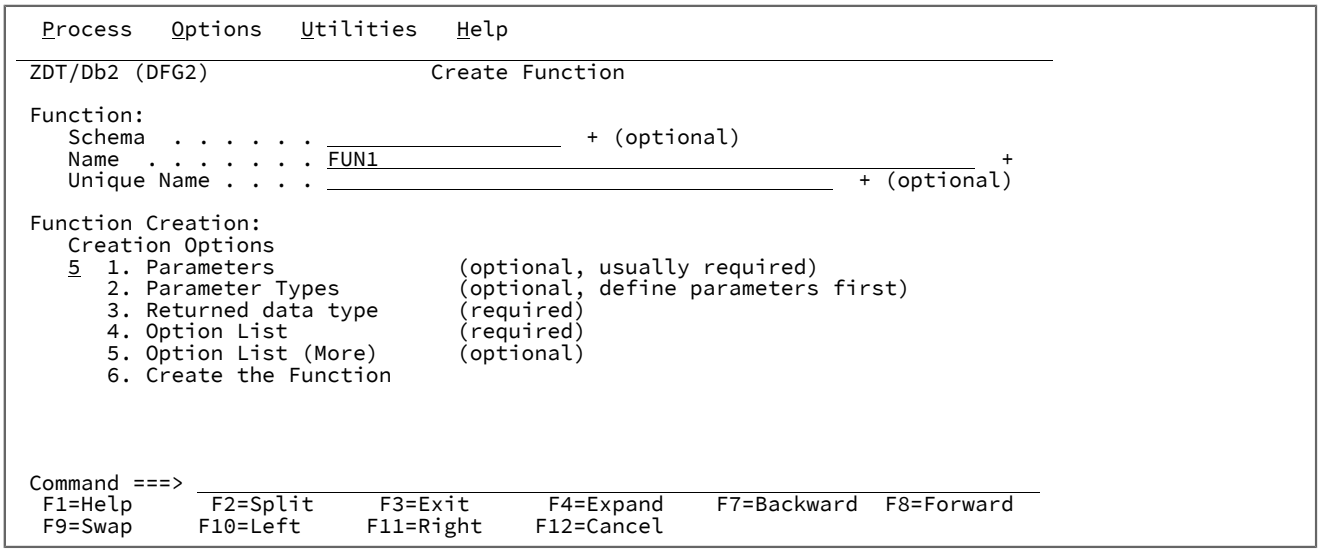
Related tasks

- Creating a function on page 239

Create Function panel

You use the Create Function panel to create a new function.

Panel and field definitions



Schema

The schema name in which to create the function.

Name

The name of the user-defined function.

Unique Name

Specifies a unique name for the function that is used by certain SQL statements and Db2® commands.

Creation Options

Specify the number of the creation option you want to use:

1. Parameters

Displays the Create Function Parameters panel for defining parameters for the function.

2. Parameter Types

Displays the Create Function Parameter Types panel for specifying additional type information for the function parameters.

3. Returned data type

Displays the Create Function: Returned Data Type panel for specifying the data type returned by the function.

4. Option List

Displays the Create Function Option List (1/2) panel for specifying various options for further defining the function.

5. Option List (More)

Displays the Create Function Option List (2/2) panel for specifying additional options for the function.

6. Create the Function

When you have finished specifying parameter, data type, and option information as required, select this creation option to create the function.

Parent panels

- Db2 Object Functions panel on page 622
- Functions panel on page 709

Child panels

To display this panel…	Use/do this
Create Function: Parameters panel on page 539	Creation option 1
Create Function: Parameter Types panel on page 537	Creation option 2
Create Function: Returned Data Type panel on page 540	Creation option 3
Create Function: Option List (1/2) panel on page 533	Creation option 4
Create Function: Option List (2/2) panel on page 534	Creation option 5

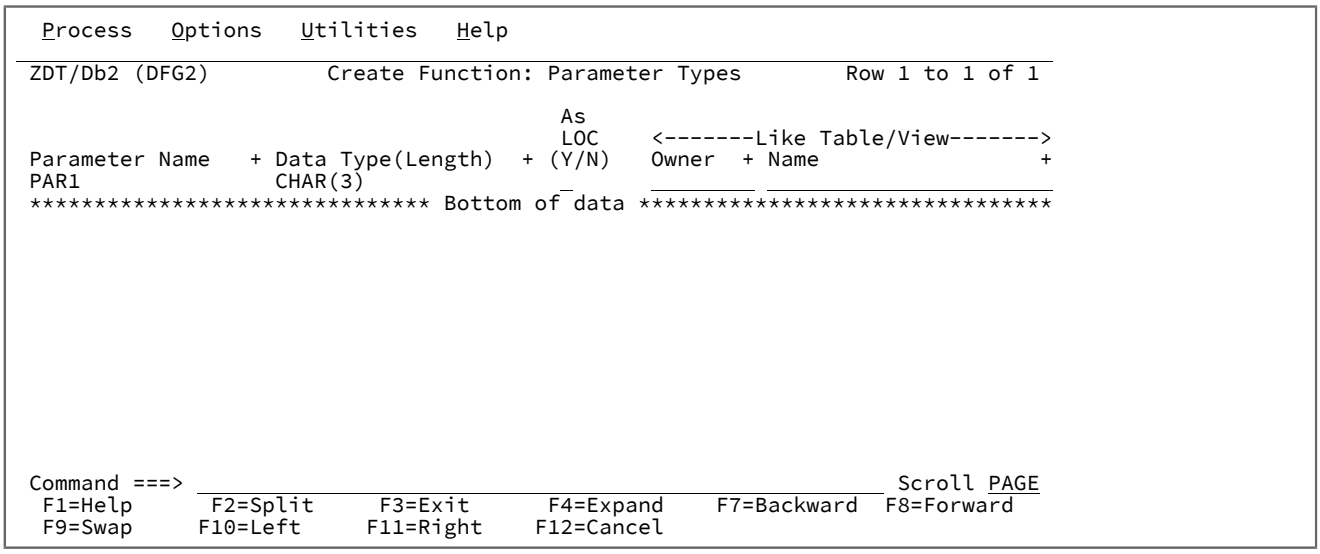
Related tasks

- Creating and dropping Db2 objects on page 208
- Creating a function on page 239

Create Function: Parameter Types panel

You use the Create Function: Parameter Types panel to specify additional type information for the function parameters.

Panel and field definitions



Parameter Name

As defined using option 1.

Data Type (Length)

As defined using option 1.

As LOC (Y/N)

For LOB data types, specifies whether a locator for the parameter is passed instead of the value itself:

Y

Locator passed

N

Value passed

Owner

Defines the parameter as a locator to a transition table.

Like Table/View Name

The owner and name specify a table or view with columns that match those of the transition table.

Parent panels

- Create Function panel on page 536

Child panels

None.

Related tasks

- Creating a function on page 239

Create Function: Parameters panel

You use the Create Function: Parameters panel to define parameters for the function.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
 ZDT/Db2 (DFG2)            Create Function Parameters          Row 1 to 1 of 1
                        <---(For Built in Data Types)----->
 C                                          <DEC>
 M                                         <FL>   For   User Defined
 D  Parameter Name   + Data Type       + Len   Pr Sc Data  Data Type          +
 __ NAME             CHAR              20    __
__ _____ ____________________
 ****************************** Bottom of data ************************
 Command ===> _____________________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
  F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Cmd

This field is used for entering commands.

Parameter Name

As defined using option 1.

Data Type

This field is used for entering the data type for the column being defined.

Len

This field is (optionally) used for entering the length, and optionally the scale of the column. The numeric fields such as INTEGER, REAL, no length specification is required. For character fields, only a single length specification is required. For decimal fields, you can specify a length and a scale.

DEC

For a decimal number. The first integer is the precision number. That is, the total number of digits which can range from 1 to 31. The second integer is the scale of the number. That is the number of digits to the right of the decimal point, which can range from 0 to the precision of the number.

FL

For a floating-point number. If integer is between 1 and 21 inclusive, the format is single precision floating-point. If the integer is between 22 and 53 inclusive, the format is double precision floating-point.

Pr Sc

The scale is the total number of digits in the fractional part of the number and the precision is the total number of digits in the number.

For Data

The definitions for character and character large object (CLOB) allow the specification of an optional 'FOR DATA' clause. This field is used to specify the appropriate value.

User Defined Data Type

Specifies the data type of the column is a distinct type (a user defined-data type). The length, precision and scale of the column are respectively the length, precision and scale of the source type of the distinct type. The privilege set must implicitly or explicitly include the USAGE privilege on the distinct type.

Parent panels

- Create Function panel on page 536

Child panels

None.

Related tasks

- Creating a function on page 239

Create Function: Returned Data Type panel

You use the Create Function: Returned Data Type panel to specify the data type returned by the function.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)       Create Function: Returned Data Type


Returns:
   Schema   . . . . . . . . . . _________________  + (optional, for user-defined types)
   Data Type  . . . . . . . . . _____________________________________________  +
   Length/Precision . . . . . . ________ (optional)
   Scale  . . . . . . . . . . . ________ (optional)
   For Data                       Encoding Method
   _ 1. SBCS                      _ 1. EBCDIC
     2. Mixed                       2. ASCII
     3. Bit                         3. UNICODE

Cast From:
   Data Type  . . . . . . . . . _____________________________________________  + (optional)
   Length/Precision . . . . . . ________ (optional)
   Scale  . . . . . . . . . . . ________ (optional)
   For Data                       Encoding Method
   _ 1. SBCS                      _ 1. EBCDIC
     2. Mixed                       2. ASCII
     3. Bit                         3. UNICODE

As Locator:
   Enter "/" to select option
      Return a locator to the value (LOB data type only)
Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Schema

Specifies the schema name that qualifies a user-defined data type.

Data Type (Returns)

Specifies the data type returned by the function.

Length/Precision (Returns)

Specifies the length for built-in data types. For the decimal data type this is the precision.

Scale (Returns)

For the DECIMAL data type only.

For Data (Returns)

1.SBCS

Only for built in data types.

2.Mixed

Only for built in data types.

3.Bit

Only for built in data types.

Encoding Method (Returns)

Only for built-in data types. Specifies the actual data type returned by the function. See note below.

Data Type (Cast From)

Specifies the data type returned by the function.

Length/Precision (Cast From)

Specifies the length for built-in data types. For the decimal data type this is the precision.

Scale (Cast From)

For the DECIMAL data type only.

For Data (Cast From)

1.SBCS

Only for built in data types.

2.Mixed

Only for built in data types.

3.Bit

Only for built in data types.

Encoding Method (Cast From)

See note below.

As Locator

For LOB data types, specifies that a locator for the parameter is returned instead of the value itself.

Note: If *"Cast From"* is not specified, then the *"Returns"* **Encoding Method** defines the data type returned by the function to the invoking statement. If *"Cast From"* is specified, then the *"Cast From"* **Encoding Method** is the data type returned by the function and the *"Returns"* **Encoding Method** is the data type passed to the invoking statement by Db2® (having *"cast"* it from the *"Cast From"* data type).

Parent panels

- Create Function panel on page 536

Child panels

None.

Related tasks

- Creating a function on page 239

Create Index panel

You use the Create Index panel to create an index for a pre-existing base table.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                   Create Index

New Index:
   Owner . . . . . . . _________________  + (optional)
   Name  . . . . . . . _____________________________________________________  +
   Table Owner . . . . _________________  + (optional)
   Table Name  . . . . _____________________________________________________  +
   Buffer Pool . . . . ____                (optional, BP0-BP49)

Index Creation:
   Creation Options
   2  1. Index type             (optional)
      2. Index columns          (required, except on auxiliary tables)
      3. Allocation             (optional, usually needed)
      4. Index options          (optional)
      5. Partitions             (partitioned indexes only)
      6. Partition values       (required for partitioned indexes)
      7. Create the index

Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Owner

The owner of the index.

Name

The name of the index. This is a required field.

Table Owner

The SQLID used to create the table on which the index is being defined.

Table Name

The name of the table on which the index is being created. This is a required field. You can enter the name of the table on which the index is created. A validation check needs to be performed to verify that the table exists.

Buffer Pool

The name of the buffer pool in which the index data sets are created.

Creation Options

Specify the number of the creation option you want to use:

1. **Index type**

   To specify information to define the type of index to be created. Displays the Create Index: Type panel.

2. **Index columns**

   To select the columns that comprise the index. Displays the Create Index: Column Selection panel.

3. Allocation

To specify the space requirements for the index being created. displays the Create Index: Allocation panel.

4. Index options

To specify information to define options for the index to be created. Displays the Create Index: Options panel.

5. Partitions

To specify space information for index partitions. displays the Create Index: Partitions panel.

6. Partition values

To specify limit values for index entries in each partition. displays the Create Index: Partition Values panel.

7. Create the index

When you have finished specifying details for the index to be created, select this creation option to create the index.

Parent panels

- Db2 Object Functions panel on page 622
- Tables, Views and Aliases panel on page 891
- Indexes panel on page 723
- Explain Utilities panel on page 686

Child panels

To display this panel...	Use/do this
Create Index: Type panel on page 552	Creation option 1
Create Index: Column Selection panel on page 546	Creation option 2
Create Index: Allocation panel on page 545	Creation option 3
Create Index: Options panel on page 547	Creation option 4
Create Index: Partitions panel on page 549	Creation option 5
Create Index: Partition Values panel on page 550	Creation option 6

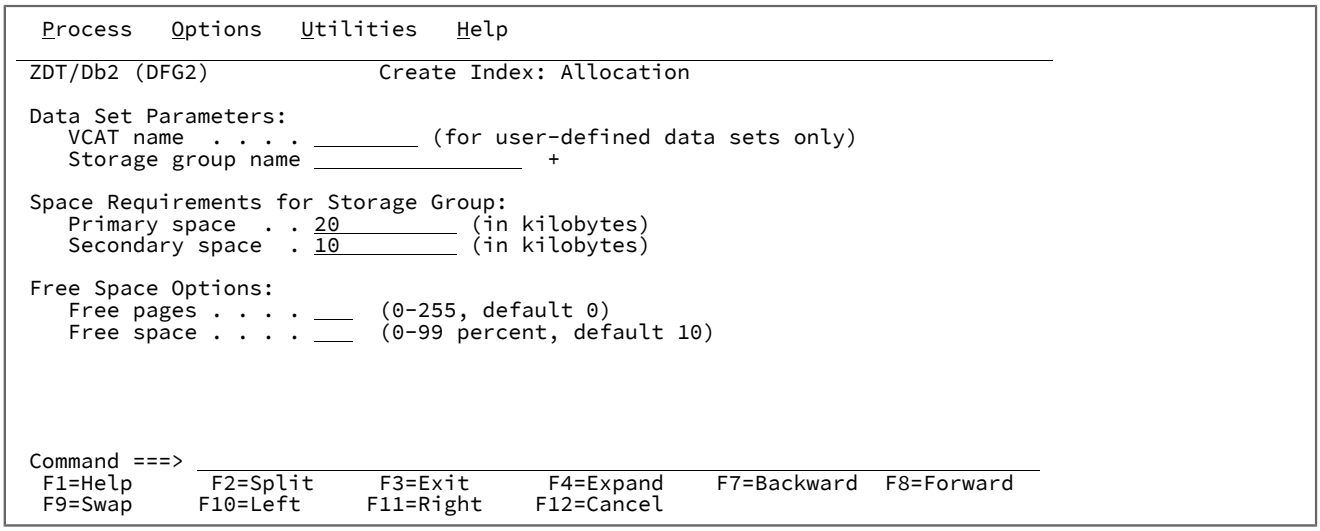
Related tasks

- Creating an index on page 236

Create Index: Allocation panel

You use the Create Index: Allocation panel to specify the space requirements for the index being created.

Panel and field definitions



VCAT name

User-defined data sets.

Storage group name

Db2-defined data sets (STOGROUP name)

Primary space

Specifies the minimum primary space allocation for a managed Db2® data set.

Secondary space

Specifies the minimum secondary space allocation for a managed Db2® data set.

Free pages

Valid values are 0-255. For a detailed description, see *DB2 for z/OS SQL Reference*.

Free space

Valid values are 0-99. For a detailed description, see *Z Data Tools User's Guide and Reference for IMS Data*.

Parent panels

- Create Index panel on page 542

Child panels

None.

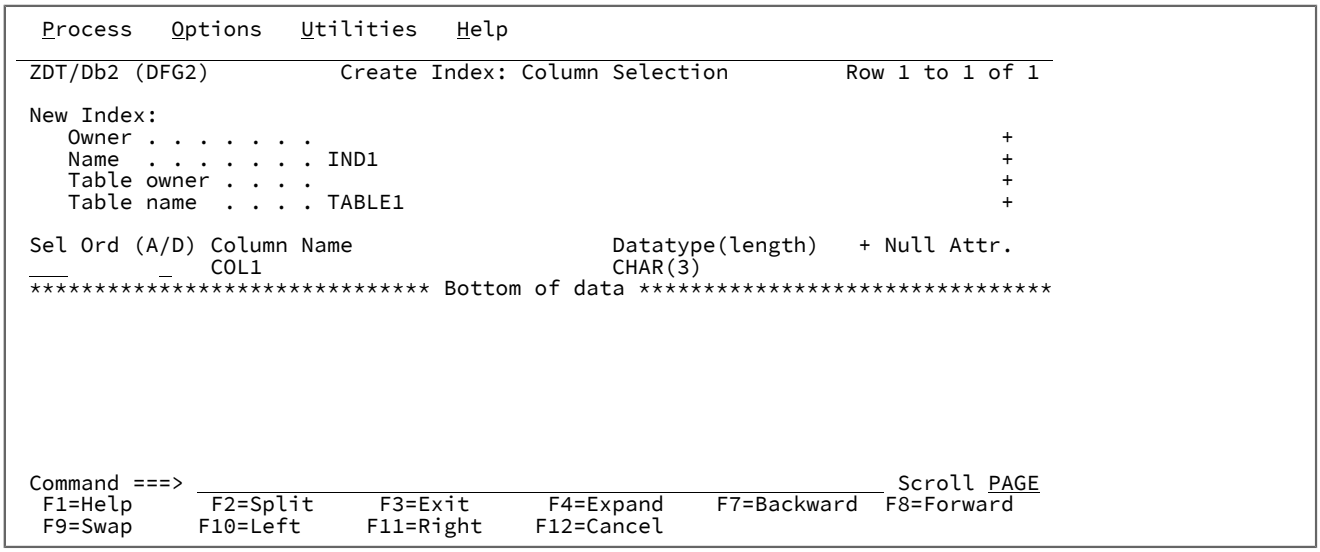
Related tasks

- Creating an index on page 236

Create Index: Column Selection panel

You use the Create Index: Column Selection panel to select the columns that comprise the index.

Panel and field definitions



Owner

The name of the owner of the new index.

Name

The name of the new index.

Table owner

The SQLID used to create the table on which the index is being defined.

Table name

The table on which the index is created.

The bottom half of the panel shows a table that can be scrolled in the usual ISPF manner.

Sel

An input selection field. This is where you identify the columns that make up the index. You can enter `s` to select or deselect any column. You can also type a number against a column, to indicate the ordering of the columns in the index.

Order

A display-only field. Possible values are spaces, indicating this column is not part of the index; or a number, indicating that this column has been selected as part of the index, and its order among the selected columns.

(A/D)

Indicates how the values in the column are to be indexed. If **Sel** is blank, but **A/D** is not blank, it is as if **Sel** contains *"S"*. The available options are:

A

Ascending. This is the default. If you do not specify a value, this field is filled with an *"A"* when the panel is redisplayed.

D

Descending

Column Name

The name of the column.

Data Type(length)

The Db2® data type of the column.

Null Attribute

Shows whether column allows null values.

Parent panels

- Create Index panel on page 542

Child panels

None.

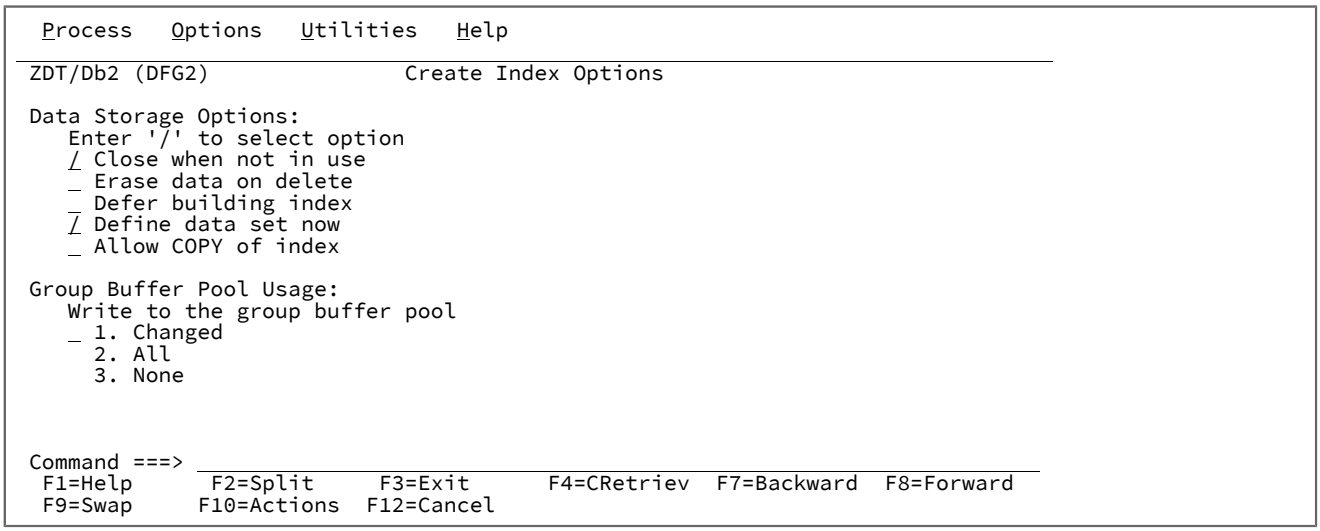
Related tasks

- Creating an index on page 236

Create Index: Options panel

You use the Create Index: Options panel to specify information to define options for the index to be created.

Panel and field definitions



You can use the first group of input fields to select the appropriate option with the '/' selection character. The following section describes the options:

Close when not in use

This corresponds to the CLOSE YES/NO clause. It relates to how Db2® handles open data sets when the limit for the maximum number of open data sets is reached.

Erase data on delete

This corresponds to the ERASE YES/NO clause in the 'USING' block. ERASE YES means that the data sets associated with the partition are overwritten with binary zeros when the partition is deleted (dropped).

Defer building index

This corresponds to the DEFER YES/NO clause. The default value is DEFER NO, meaning that the index is built when the CREATE INDEX statement is executed.

Define data set now

This corresponds to the DEFINE YES/NO clause. The default value is DEFINE YES, meaning that the data sets used for the index are created when the CREATE INDEX statement is executed.

Allow COPY of index

This corresponds to the COPY YES/NO clause. The default value is COPY NO, meaning that it is not possible to use the COPY utility against the index.

The second input field on the panel contains three options relating to the use of the grouped buffer pool. You can select the required option by typing the appropriate number. This corresponds to the GBPCACHE clause in the create statement syntax.

Parent panels

- Create Index panel on page 542

Child panels

None.

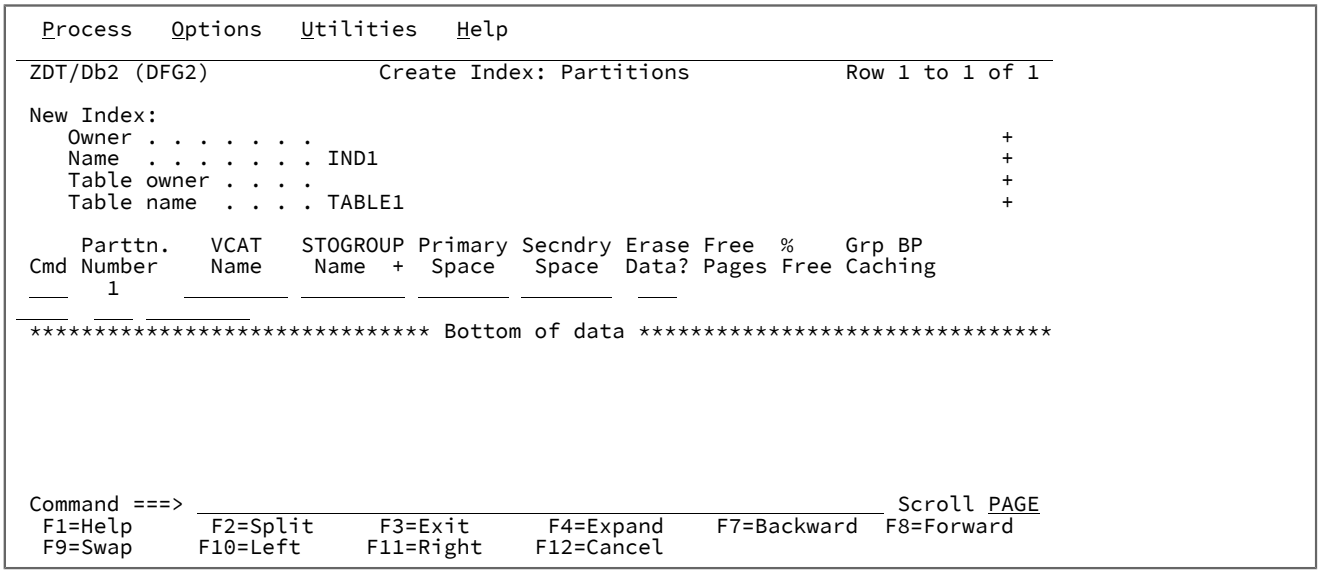
Related tasks

- Creating an index on page 236

Create Index: Partitions panel

You use the Create Index: Partitions panel to specify space information for index partitions.

Panel and field definitions



Owner

The name of the owner of the new index.

Table owner

The SQLID used to create the table on which the index is being defined.

Name

The name of the new index.

Table name

The table on which the index is created.

Cmd

An input selection field. When the panel is first displayed, the table contains 1 row, with default values entered. To define additional partitions type 'I' in the selection field and press Enter. You can also type 'R' to repeat the current definition. Other valid selection commands are i and r followed by an integer.

Parttn. Number

A sequential number showing the number of partitions defined. This is a display only field.

VCAT Name

The VCAT name used to define user-specified data sets. This is an input field. It is mutually exclusive with the STOGROUP parameter.

STOGROUP name

The Db2® storage group name in which the index data set is defined. This is an input field. It is mutually exclusive with the VCAT parameter.

Primary Space

The primary space allocation for the partition, entered in kilobytes.

Secndry Space

The secondary space allocation for the partition, entered in kilobytes.

Erase Data?

A YES/NO value corresponding to the ERASE YES/NO clause.

Free Pages

Free pages results in Db2® leaving an empty page every *n* pages (*n* is what is specified).

% Free

The value specified here results in Db2® leaving *n*% (*n* is what is specified) of each page free.

Grp BP Caching

This corresponds to the GBPCACHE clause of the create statement (for each partition). These options only apply in a data sharing environment.

Parent panels

- Create Index panel on page 542

Child panels

None.

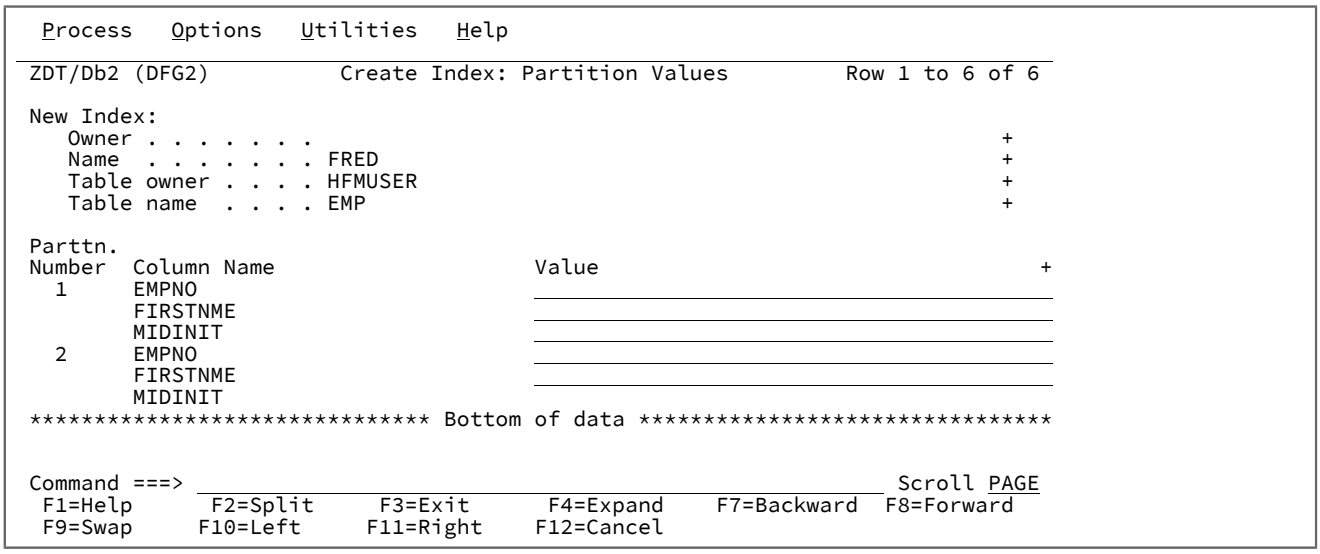
Related tasks

- Creating an index on page 236

Create Index: Partition Values panel

You use the Create Index: Partition Values panel to specify limit values for index entries in each partition.

Panel and field definitions



Owner

The name of the owner of the new index.

Name

The name of the new index.

Table owner

The SQLID used to create the table on which the index is being defined.

Table name

The table on which the index is created.

Parttn. Number

A sequential number showing the number of partitions defined. This is a display-only field.

Column Name

The name of the column.

Value

The smallest unit of data that is manipulated in SQL.

Parent panels

- Create Index panel on page 542

Child panels

None.

Related tasks

- Creating an index on page 236

Create Index: Type panel

You use the Create Index: Type panel to specify information to define the type of index to be created.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)               Create Index Type

Index Type Options:
   Uniqueness
   1 1. Non-Unique
     2. Unique
     3. Unique Not Null

Clustering Index Option:
   Enter '/' to select option
   / Clustering Index (required for partitioned indexes)

Non-Partitioned Index:
   Piecesize  . . . . ________     Multiplier
                                __ 1. K (1024)
                                   2. M (1048576)
                                   3. G (1073741824)



(Specify the partitions for a partitioned index using Create Index option 5)
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

1. Non-Unique

Duplicate index values are permitted.

2. Unique

Distinct values for all entries in the index.

3. Unique Not Null

Distinct values for all not null entries in the index.

Clustering Index Option

Enter a "/" to specify a clustering index (CLUSTER keyword).

Piecesize

You can specify a maximum addressability for each piece (data set of a non-partitioning index). The values entered must be powers of 2, and the valid values depend on the multiplier.

- If the multiplier is K, valid values are 2**n between 256 and 67 108 864 inclusive.
- If the multiplier is M, valid values are 2**n between 1 and 66 536 inclusive.
- If the multiplier is G, valid values are 2**n between 1 and 64 inclusive.

Parent panels

- Create Index panel on page 542

Child panels

None.

Related tasks

- Creating an index on page 236

Create Procedure panel

You use the Create Procedure panel to create a new procedure.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                 Create Procedure


Stored Procedure:
Schema  . . . _________________  + (optional)
Name  . . . . ____________________________________________________  +
Result Sets . _________ (optional)


Stored Procedure Creation:
   Creation Options
   _  1. Parameters              (optional, usually required)
      2. Parameter Types         (optional, define parameters first)
      3. Option List             (required)
      4. Option List (More)      (optional)
      5. Create the Procedure

Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Schema

The name of the schema in which the procedure is defined.

Name

The name of the stored procedure.

Result Sets

Specifies the maximum number of dynamic result sets that can be returned by the stored procedure.

Creation Options

Specify the number of the creation option you want to use:

1. **Parameters**

   To specify information to define the parameters for a procedure to be created. Displays the Create Procedure: Parameters panel.

2. **Parameter Types**

   To specify information to define the parameter types in relation to tables or views, and to specify the parameter type. Displays the Create Procedure: Parameter Types panel.

3. **Option List**

   To specify information to define options for the procedure to be created. Displays the Create Procedure: Option List (1/2) panel.

4. **Option List (More)**

   To further specify information to define options for the procedure to be created. Displays the Create Procedure: Option List (2/2) panel.

5. **Create the Procedure**

   When you have finished specifying details for the stored procedure to be created, select this creation option to create the stored procedure.

Parent panels

- Db2 Object Functions panel on page 622
- Stored Procedures panel on page 852

Child panels

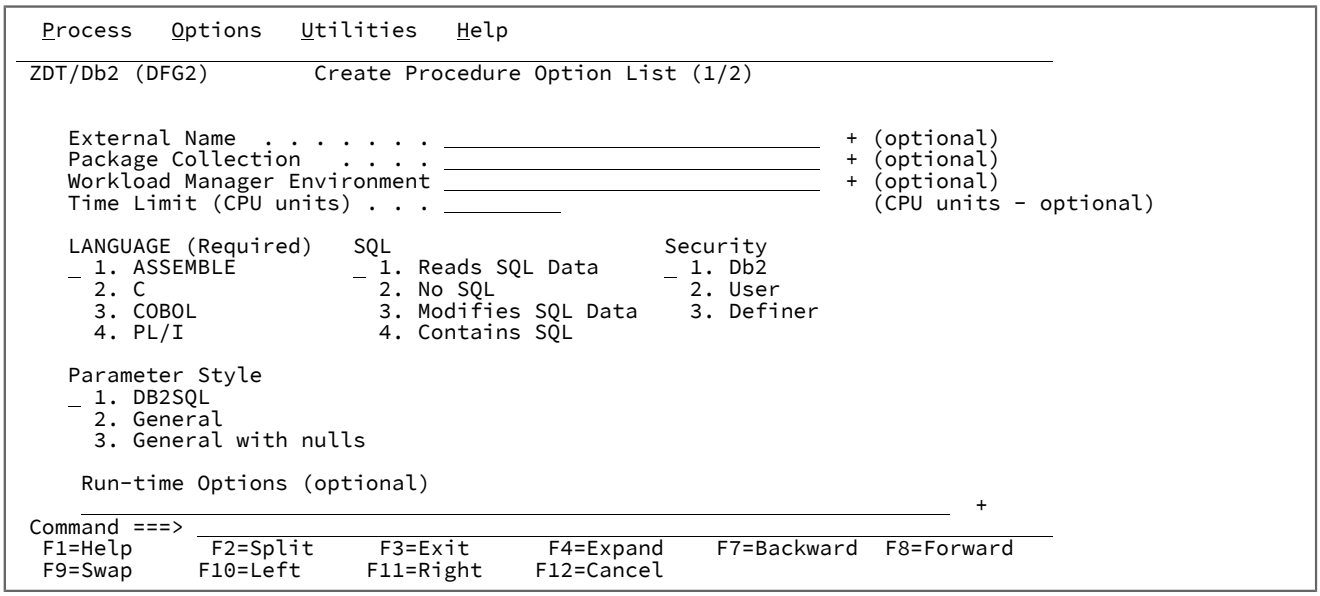
To display this panel…	Use/do this
Create Procedure: Parameters panel on page 558	Creation option 1
Create Procedure: Parameter Types panel on page 557	Creation option 2
Create Procedure: Option List (1/2) panel on page 554	Creation option 3
Create Procedure: Option List (2/2) panel on page 556	Creation option 4

Related tasks

Create Procedure: Option List (1/2) panel

You use the Create Procedure: Option List (1/2) panel to specify information to define options for the procedure to be created.

Panel and field definitions



External Name

Name of the MVS™ load module that contains the user-written code to implement the procedure.

Package Collection

Specifies the package collection to be used when the procedure is run. If not specified, the invoking program's package collection is used.

Workload Manager Environment

Specifies the MVS™ workload manager environment in which the procedure is to run.

Time Limit (CPU units)

Limit on processor service units used by a single invocation of the procedure.

Language

Application programming language of the procedure.

SQL

Specifies what type of SQL statements the procedure contains.

Security

Specifies the interaction of the procedure with an external security product.

Parameter Style

1. DB2SQL

Additional parameters containing Db2® information are passed to the procedure.

2. General

Only the parameters on the call statement are passed to the procedure and they cannot be null.

3. General with nulls

A vector of null indicators is passed in addition to the parameters on the call statement.

Parent panels

- Create Procedure panel on page 553

Child panels

None.

Related tasks

- Creating a procedure on page 240

Create Procedure: Option List (2/2) panel

You use the Create Procedure: Option List (2/2) panel to further specify information to define options for the procedure to be created.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)        Create Procedure Option List (2/2)


Enter '/' to select option
_  Main Program
_  Stay Resident
_  Deterministic
_  DBINFO argument passed
_  Commit on return







Command ===>                                                  Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Main Program

If checked, the program runs as a main routine, otherwise it runs as a subroutine.

Stay Resident

If checked, the function load module remains in memory when the function ends.

Deterministic

If checked, indicates the function always returns the same result for the same set of input values.

DBINFO argument passed

If checked, indicates that additional Db2® information is passed when the function is called.

Commit on return

If checked, indicates that Db2® issues a commit when the stored procedure returns.

Parent panels

- Create Procedure panel on page 553

Child panels

None.

Related tasks

- Creating a procedure on page 240

Create Procedure: Parameter Types panel

You use the Create Procedure: Parameter Types panel to specify information to define the parameter types in relation to tables or views, and to specify the parameter type.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)        Create Procedure: Parameter Types        Row 1 to 1 of 1

                                  As                                    IN
                    Data Type(Length)  LOC  <------Like Table/View------> OUT
Parameter Name   +                  + (Y/N) Owner  + Name               + INOUT
PARAM1              CHAR(3)          _       ________ ___________________ _____
****************************** Bottom of data ********************************









Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Parameter Name

The name of the parameter.

Data Type(Length)

The data type of the parameter.

As LOC (Y/N)

For LOB data types, specifies that a locator for the parameter is passed instead of the value itself.

Owner

Defines the parameter as a locator to a transition table.

Like Table/View Name

The owner and name specify a table or view with columns that match those of the transition table.

IN OUT INOUT

Specifies whether the parameter can be used for input only, output only, or both input and output. Valid values are:

IN

Identifies the parameter as an input parameter.

OUT

Identifies the parameter as an output parameter.

INOUT

Identifies the parameter as both an input and output parameter.

Parent panels

- Create Procedure panel on page 553

Child panels

None.

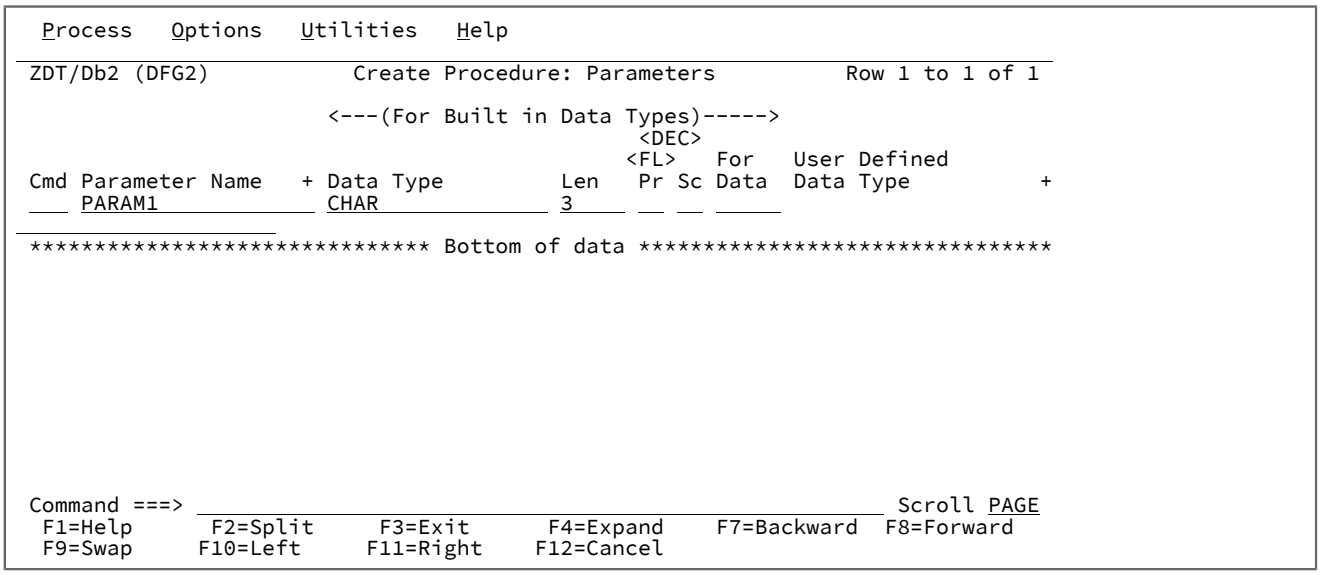
Related tasks

- Creating a procedure on page 240

Create Procedure: Parameters panel

You use the Create Procedure: Parameters panel to specify information to define the parameters for a procedure to be created.

Panel and field definitions



Cmd

This field is used for entering commands.

Parameter Name

As defined using option 1.

Data Type

This field is used for entering the data type for the column being defined.

Len

This field is (optionally) used for entering the length, and optionally the scale of the column. The numeric fields such as INTEGER, REAL, no length specification is required. For character fields, only a single length specification is required. For decimal fields, you can specify a length and a scale.

Dec

For a decimal number. The first integer is the precision number. That is, the total number of digits which can range from 1 to 31. The second integer is the scale of the number. That is the number of digits to the right of the decimal point, which can range from 0 to the precision of the number.

FL

For a floating-point number. If integer is between 1 and 21 inclusive, the format is single precision floating-point. If the integer is between 22 and 53 inclusive, the format is double precision floating-point.

Pr Sc

The scale is the total number of digits in the fractional part of the number and the precision is the total number of digits in the number.

For Data

The definitions for character and character large object (CLOB) allow the specification of an optional 'FOR DATA' clause. This field is used to specify the appropriate value.

User Defined Data Type

Specifies the data type of the column is a distinct type (a user defined-data type). The length, precision and scale of the column are respectively the length, precision and scale of the source type of the distinct type. The privilege set must implicitly or explicitly include the USAGE privilege on the distinct type.

Parent panels

- Create Procedure panel on page 553

Child panels

None.

Related tasks

- Creating a procedure on page 240

Create Synonym panel

You use the Create Synonym panel to create a new synonym.

Panel and field definitions

```
  Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
 ZDT/Db2 (DFG2)                  Create Synonym

 Synonym Details:
 Name . . . . . . . .  DBNAME                                            +

 Table or View Details:
 Owner  . . . . . . .  ________________  +
 Name . . . . . . . .  TABLE1                                            +









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Name (Synonym Details)

The name of the synonym that is being created.

Owner

The owner of the table.

Name (Table or View Details)

The name of the table.

Parent panels

- Db2 Object Functions panel on page 622
- Synonyms panel on page 855

Child panels

None.

Related tasks

- Creating a synonym on page 238

Create Table panel

You use the Create Table panel to create a new table.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                   Create Table

New Table:
   Owner . . . . . . . _________________  + (optional)
   Name  . . . . . . . TABLE1_____________________________________________  +
   Database  . . . . . _________            (optional)
   Table Space . . . . _________            (optional)

Model Table/View:  (optional)
   Owner . . . . . . . _________________  +
   Name  . . . . . . . _____________________________________________________  +
   Usage:
      1  1. Generate LIKE clause
         2. Load table information

Table Creation:
   Creation Options              (Use option 1 before options 2, 4-9)
   1  1. Columns                 (normally required, invalid with model LIKE)
      2. Nulls/default values    (optional)
      3. Table options           (editproc,validproc,more; optional)
      4. Unique Constraints      (primary/unique keys; optional)
      5. Referential Constraints (foreign keys; optional)
      6. Check constraints       (optional)
      7. Procedure exits         (optional)
      8. Generate values         (optional)
      9. Create the table
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Owner (New Table)

The name of the owner of the new table.

Name (New Table)

The name of the new table.

Database

The name of the database in which the new table is to be created.

Table Space

The name of the table space in which the new table is to be created.

Owner (Model Table/View)

The name of the owner of the model table.

Name (Model Table/View)

The name of the model table.

Creation Options

Specify the number of the creation option you want to use:

1. **Columns**

   To specify information to define the columns for a table to be created. Displays the Create Table: Columns panel.

2. **Nulls/default values**

   To specify information to define the null attributes and default values for the columns of the table. Displays the Create Table: Nulls and Default Values panel.

3. **Table options**

   To specify information to define options for the table you are creating. Displays the Create Table: Options panel.

4. **Unique Constraints**

   To specify information to define a primary key and up to three additional unique keys for the table you are creating. Displays the Create Table: Unique Constraints panel.

5. **Referential Constraints**

   To specify information to define one or more column referential constraints for the table. Displays the Create Table: Column Referential Constraints panel.

6. **Check Constraints**

   To specify information to define column check constraints for the table. Displays either the Create Table: Column Check Constraints panel or the Create Table: Table Check Constraints panel.

7. **Procedure exits**

   To specify information to define procedure exits for the table you are creating. Displays the Create Table: Procedure Exits panel.

8. Generate values

To specify information to define generated values for identity or ROWID columns in the table. Displays the Create Table: Generate Values panel.

9. Create the table

When you have finished specifying details for the table to be created, select this creation option to create the table.

Parent panels

- Db2 Object Functions panel on page 622
- Table Spaces panel on page 879
- Tables, Views and Aliases panel on page 891

Child panels

To display this panel…	Use/do this
Create Table: Columns panel on page 578	Creation option 1
Create Table: Nulls and Default Values panel on page 584	Creation option 2
Create Table: Options panel on page 587	Creation option 3
Create Table: Unique Constraints panel on page 595	Creation option 4
Create Table: Column Referential Constraints panel on page 576	Creation option 5[24]
Create Table: Column Check Constraints panel on page 575	Creation option 6
Create Table: Procedure Exits panel on page 589	Creation option 7
Create Table: Generate Values panel on page 580	Creation option 8
Create Table: Model Load panel on page 582	[25]
Create Table: Columns panel on page 578	[24]

24. When you select Table Creation option 5 (**Referential Constraints**) on the main Create Table panel, ZDT/Db2 displays either the Column Referential Constraints panel (Create Table: Column Referential Constraints panel on page 576) or the Table Referential Constraints panel (Create Table: Table Referential Constraint panel on page 592). To change from one panel to the other, press the F5 function key.
25. The Create Table Model Load panel is displayed when you:

    - Initially specify, or subsequently change, the **Model Table/View** details,
    - Select Usage option 2 (**Load table information**), and
    - Press Enter.

Related tasks

- Creating a table on page 226

Create Table Space panel

You use the Create Table Space panel to create a new table space.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                Create Table Space

Table Space Definition:
   Name . . . . . . . . . . . _________ (required)
   Database . . . . . . . . . _________ (optional)
   Buffer pool  . . . . . . . _________ (optional: BPn,BP8Kn,BP16Kn,BP32Kn)

Table Space Creation:
   Creation Options
   1  1. Table space type               (optional, usually needed)
      2. Table space allocation         (optional, usually needed)
      3. Data storage options           (optional)
      4. Locking options                (optional)
      5. Data sharing options           (optional)
      6. Partitions                     (partitioned table spaces only)
      7. Create the table space


Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Name

The name of the table space you want to create.

Database

The name of the database in which the table space is to be created.

Buffer pool

The name of the buffer pool to be used for the table space.

Creation Options

Specify the number of the creation option you want to use:

1. Table space type

To specify information to define the type of table space to be created. Displays the Create Table Space: Type panel.

2. Table space allocation

To specify information to define the data set and space allocation for the table space to be created. Displays the Create Table Space: Allocation panel.

3. Data storage options

To specify information to define data storage options for the table space to be created. Displays the Create Table Space: Data Storage Options panel.

4. Locking options

To specify information to define the locking options for the table space to be created. Displays the Create Table Space: Locking Options panel.

5. Data sharing options

To specify information to define the data sharing options for the table space to be created. Displays the Create Table Space: Data Sharing Options panel.

6. Partitions

To specify information to define the partitions for a partitioned table space to be created. Displays the Create Table Space: Define Partitions panel.

7. Create the table space

When you have finished specifying details for the table space to be created, select this creation option to create the table space.

Parent panels

- Db2 Object Functions panel on page 622
- Databases panel on page 609
- Table Spaces panel on page 879

Child panels

To display this panel…	Use/do this
Create Table Space: Type panel on page 573	Creation option 1
Create Table Space: Allocation panel on page 566	Creation option 2
Create Table Space: Data Storage Options panel on page 568	Creation option 3
Create Table Space: Locking Options panel on page 572	Creation option 4
Create Table Space: Data Sharing Options panel on page 567	Creation option 5
Create Table Space: Define Partitions panel on page 570	Creation option 6

Related tasks

- Creating and dropping Db2 objects on page 208
- Creating a table space on page 224

Create Table Space: Allocation panel

You use the Create Table Space: Allocation panel to specify information to define the data set and space allocation for the table space to be created.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)          Create Table Space: Allocation

Dataset Parameters:
VCAT name . . . . . .  _________    (for user defined data sets only)
Storage group name  . .  _________  + (only if VCAT name omitted)

Space Requirements (Only if Storage Group Name specified):
   Primary space  . . .  ________   KB
   Secondary space  . .  ________   KB

Free Space and Rows per Data Page:
   Free pages . . . . .  ___        (0,255 inclusive, default value is 0)
   Free space . . . . .  __         (0-99 inclusive, default value is 5)
   Maximum rows per page ___        (1-255 inclusive, default value is 255)



Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

VCAT name

Specifies the first data set for the table space is managed by the user.

Storage group name

Specifies Db2® defines and manages the data sets for the table space.

Primary space

Specifies the minimum primary space allocation for a Db2® managed data set.

Secondary space

Specifies the minimum secondary space allocation for a Db2® managed data set.

Free pages

Specifies how often to leave a page of free space when the table space or partition is loaded or reorganized.

Free space

The total amount of unused space in a page.

Maximum rows per page

Specifies the maximum number of rows that Db2® considers placing on each data page.

Parent panels

- Create Table Space panel on page 564

Child panels

None.

Related tasks

- Creating a table space on page 224

Create Table Space: Data Sharing Options panel

You use the Create Table Space: Data Sharing Options panel to specify information to define the data sharing options for the table space to be created.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)     Create Table Space: Data Sharing Options

Selective Partition Locking:
    Enter '/' to select option
    _  Use selective partition locking

Group Buffer Pool Usage:
    Write to the Group Buffer Pool
    _  1. Changed
       2. All
       3. System
       4. None




Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

You can select the following option by entering a "/" in the selection field at the top of the panel:

Selective Partition Locking

See *DB2 for z/OS SQL Reference* for a detailed explanation of selective partition locking, and the software prerequisites.

Group Buffer Pool Usage

1. Changed

When there is inter-Db2 R/W interest on the table space or partition updated pages are written to the group buffer pool. When there is no inter-Db2 R/W interest, the group buffer pool is not used.

2. All

Indicates that pages are to be stored in a cache in the group buffer pool as they are read in from DASD.

3. System

Indicates that only changed system pages within the LOB table space are to be stored in a cache in the group buffer pool.

4. None

Indicates that no pages are to be stored in a cache in the group buffer pool.

Parent panels

- Create Table Space panel on page 564

Child panels

None.

Related tasks

- Creating a table space on page 224

Create Table Space: Data Storage Options panel

You use the Create Table Space: Data Storage Options panel to specify information to define data storage options for the table space to be created.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)     Create Table Space: Data Storage Options

Data Storage Options:
    Enter '/' to select option
    _  Use data compression
    /  Close when not in use
    _  Erase data on delete
    /  Define dataset now
    _  INSERT uses clustering index
    /  Track modified pages
    /  Log changes to LOB columns

Encoding Method:
    Data Encoding
    _  1. EBCDIC
       2. ASCII


Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Use data compression

This corresponds to the COMPRESS YES/NO clause. COMPRESS YES means that Db2® stores the data in compressed format.

Close when not in use

This corresponds to the CLOSE YES/NO clause. It relates to how Db2® handles open data sets when the limit for the maximum number of open data sets is reached.

Erase data on delete

This corresponds to the ERASE YES/NO clause of the 'USING' block (see *DB2 for z/OS SQL Reference*). ERASE YES means that Db2® data sets are overwritten with binary zeros when they are dropped (deleted).

Define dataset now

This corresponds to the DEFINE YES/NO clause. DEFINE YES means that Db2® defines the data sets for the table space when the create statement is executed. DEFINE NO means creation is delayed until data is placed in the table space.

INSERT uses clustering index

This corresponds to the MEMBER CLUSTER clause.

Track modified pages

This corresponds to the TRACKMOD YES/NO clause. It relates to a feature where changes to data pages are recorded in the table space map pages. The default setting is TRACKMOD YES. There are restrictions on this clause when LOB is specified.

Log changes to LOB columns

This corresponds to the LOG YES/NO clause. LOG YES can only be specified for a LOB table space.

Data encoding

Covers the encoding method option. For OS/390® systems, the default format is EBCDIC .

EBCDIC

Specifies the encoding scheme.

ASCII

Specifies the encoding scheme.

UNICODE

Specifies the encoding scheme.

Parent panels

- Create Table Space panel on page 564

Child panels

None.

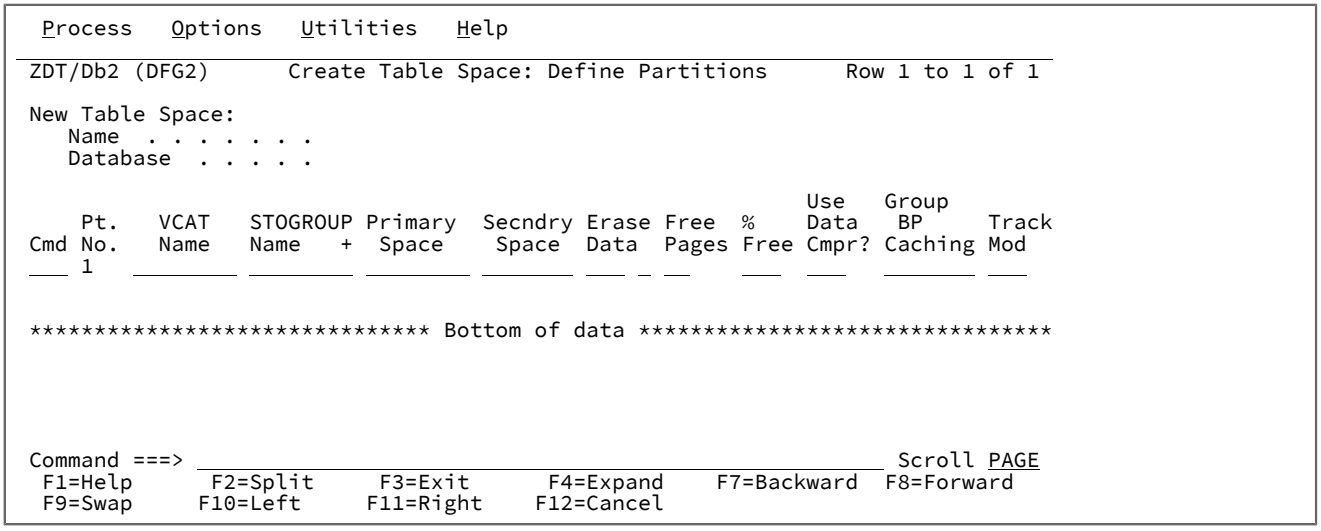
Related tasks

- Creating a table space on page 224

Create Table Space: Define Partitions panel

You use the Create Table Space: Define Partitions panel to specify information to define the partitions for a partitioned table space to be created.

Panel and field definitions



Cmd

Use the following line commands to manipulate the panel rows containing partition information:

I

Insert one new row.

I*n*

Insert *n* new rows.

R

Repeat the current row.

R*n*

Repeat the current row *n* times.

D

Delete one row.

D*n*

Delete *n* rows.

Pt. No.

Partition number

VCAT Name

This is used to specify user-defined data sets. It is mutually exclusive with a STOGROUP name.

STOGROUP Name

This is used to specify a (Db2®) storage group name. Specifying a storage group name means that Db2® creates the data sets for the partition. It is mutually exclusive with a VCAT name.

Primary Space

This is used to specify the primary space allocation for the partition (in kilobytes).

Secondary Space

This is used to specify the secondary space allocation for the partition (in kilobytes).

Erase Data

This corresponds to the ERASE YES/NO clause in the 'USING' block. ERASE YES means that the data sets associated with the partition are overwritten with binary zeros when the partition is deleted (dropped).

Free Pages

Free pages results in Db2® leaving an empty page every *n* pages (*n* is what is specified).

% Free

The value specified here results in Db2® leaving *n*% (*n* is what is specified) of each page free.

Use Data Cmpr?

This corresponds to the COMPRESS NO/YES clause. YES means the data is stored in compressed format.

Group BP Caching

This corresponds to the GBPCACHE clause of the create statement (for each partition). This option only applies in a data sharing environment.

TRACK MOD

Specifies whether Db2® tracks modified pages in the space map pages of the table space or partition. Do not specify TRACKMOD for a LOB table space. For a table space in a TEMP database, Db2® uses TRACKMOD NO regardless of the value specified.

Parent panels

- Create Table Space panel on page 564

Child panels

None.

Related tasks

- Creating a table space on page 224

Create Table Space: Locking Options panel

You use the Create Table Space: Locking Options panel to specify information to define the locking options for the table space to be created.

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)        Create Table Space: Locking Options

Locking Options:
    Locksize parameter
    _  1. ANY                      Db2 determines the appropriate lock size
       2. Table space
       3. Table                    segmented table spaces only
       4. Page                     not for LOB table spaces
       5. Row                      not for LOB table spaces
       6. LOB                      only for LOB table spaces

Application Locking:
     Maximum Locks . . . . __________ (SYSTEM (default) or 0-2147483647)




Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Locksize parameter

Specify the LOCKSIZE parameter:

1. **ANY**

   Specifies Db2® can use any lock size.

2. **Table space**

   Specifies Db2® can use table space lock.

3. **Table**

   Specifies Db2® can use table locks.

4. **Page**

   Specifies Db2® can use page locks.

5. **Row**

   Specifies Db2® can use row locks.

6. **LOB**

   Specifies Db2® can use LOB locks.

Maximum Locks

The maximum number of page, row, or LOB locks an application process can hold simultaneously in the table space. Valid values are: 0–2147483647 or *"SYSTEM"*.

Parent panels

- Create Table Space panel on page 564

Child panels

None.

Related tasks

- Creating a table space on page 224

Create Table Space: Type panel

You use the Create Table Space: Type panel to specify information to define the type of table space to be created.

Panel and field definitions

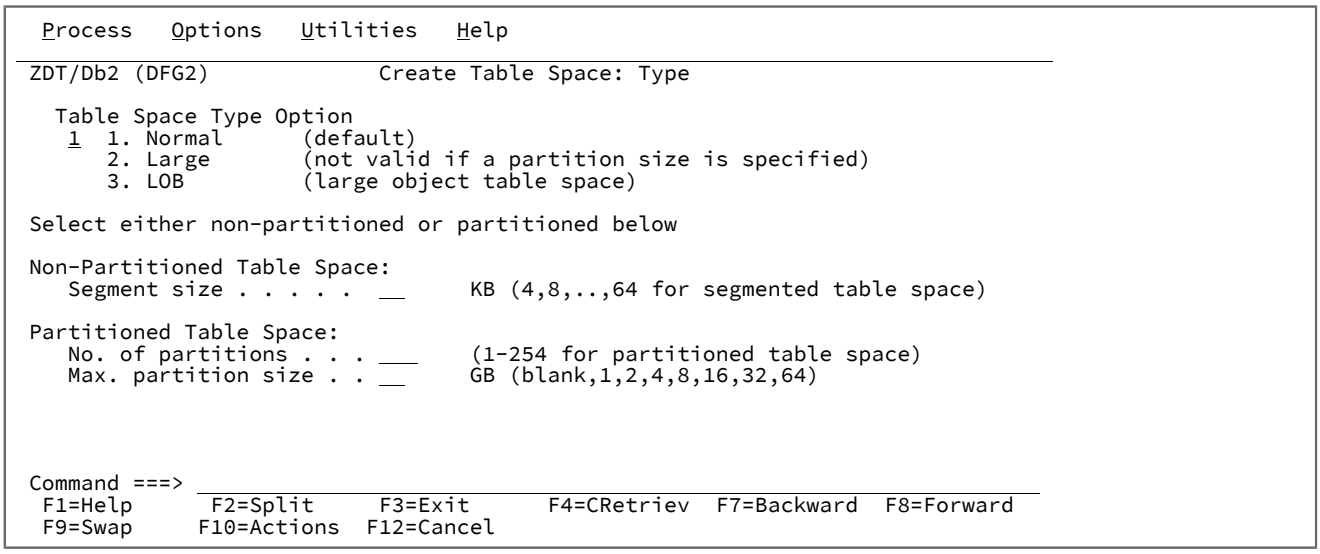


Table Space Type Option

Specify the type option:

1. Normal

Identifies the table space is a normal (non-LOB) table space.

2. Large

Identifies that each partition of a partitioned table space has a maximum partition size of 4 GB. This relates to a deprecated SQL option, therefore it is preferable to use Max. Partition Size. If **Large** is specified, then **No. of partitions** must also be specified.

3. LOB (large object)

Identifies the table space as LOB table space. If LOB is specified, neither **Segment size** or **No. of partitions** can be specified.

LOB means Large Object. A LOB table space is used to store data that does not fit within a single table space page. Examples of LOBs might include such things as photographs, video files, or sound files.

Segment size

Indicates that the table space is separated and specifies how many pages are in each segment. This cannot be specified with Large, LOB or either of the partitioned table space options.

No. of partitions

Indicates that the table space is partitioned and specifies the number of partitions. This is required if Large is specified or if Max. partition size is specified with LOB or segment size.

Max. partition size

Indicates that the maximum size of each partition or for LOB table spaces, the maximum size of each data set. This requires the specification of LOB or No. of partitions. This cannot be specified with Large or Segment size.

Parent panels

- Create Table Space panel on page 564

Child panels

None.

Related tasks

- Creating a table space on page 224

Create Table: Column Check Constraints panel

You use the Create Table: Column Check Constraints panel to specify information to define column check constraints for the table.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)      Create Table: Column Check Constraints     Row 1 to 1 of 1

   Owner . . . . . .                                                        +
   Name  . . . . . . TAB1                                                   +
   Database  . . . . .
   Table Space . . . .

                                                          Check
Sel Column Name      + Data Type(Length)  + Constraint Name  + Condition      +
 _   COL1              CHAR(3)
****************************** Bottom of data ********************************









Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Tabcons   F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Sel

Type `s` in this field to display a scrollable panel that use to specify a check condition up to the maximum length permitted by Db2®. See the description for the **Check Condition** entry field.

Column Name

The name of the column in the table you are creating.

Data Type(Length)

The data type and length of the column in the table you are creating.

Constraint Name

The name of the constraint. This is an optional field.

Check Condition

SQL expression defining the column check constraint.

If you need to specify a check condition that is too long to fit in the entry field on the panel, enter `s` in the **Sel** field to display a panel. If a long check condition already exists and you alter the part of the check condition displayed on the Column Check Constraints panel, ZDT/Db2 displays the same panel that you can use to confirm the changes you have made.



Note: ZDT/Db2 does not validate the check condition you specify.

Parent panels

- Create Table panel on page 561

Child panels

None.

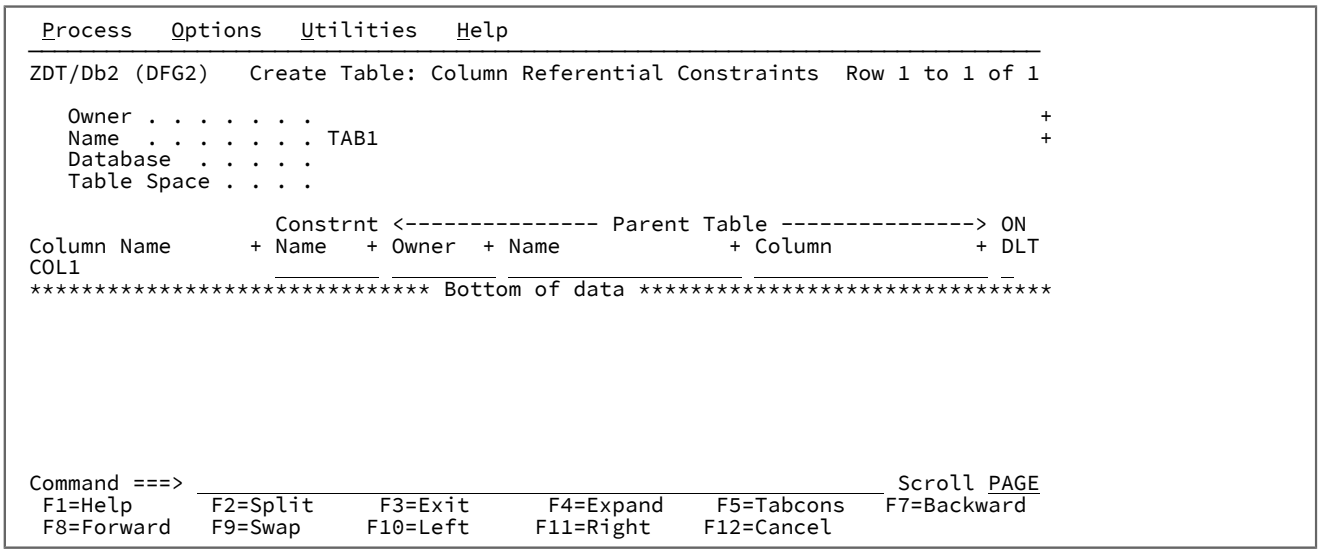
Related tasks

- Creating a table on page 226

Create Table: Column Referential Constraints panel

You use the Create Table: Column Referential Constraints panel to specify information to define one or more column referential constraints for the table.

Panel and field definitions



Column Name

The foreign key. This is the column in the table being created that is to be related to an equivalent column (the parent key) in the parent table.

Constrnt Name

Optionally use this field to specify the name of the constraint.

Owner

Optionally use this field to qualify the name of the parent table.

Name

The name of the table containing the parent key. This field is mandatory.

Column

The parent key. Specify the name of the column in the parent table to which you want the foreign key (in **Column Name**) related.

The parent key must be a primary or unique key on the parent table. If you leave the **Parent Table Column** field blank, the foreign key is related to the primary key of the parent table.

ON DLT

Determines the type of ON DELETE clause, if any, that ZDT/Db2 generates in the CREATE TABLE statement. Valid values are:

(blank)

No ON DELETE clause generated

R

ON DELETE RESTRICT

N

ON DELETE NO ACTION

C

ON DELETE CASCADE

S

ON DELETE SET NULL

Parent panels

- Create Table panel on page 561

Child panels

None.

Related tasks

- Creating a table on page 226

Create Table: Columns panel

You use the Create Table: Columns panel to specify information to define the columns for a table to be created.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)              Create Table: Columns              Row 1 to 1 of 1

   Owner . . . . . .                                                        +
   Name  . . . . . . TAB1                                                   +
   Database  . . . . .
   Table Space . . . .

                       <---(For Built in Data Types)----->
                                                <DEC>
                                               <FL>   For   User-defined
Cmd Column Name      + Data Type          Len    Pr Sc Data  Data Type          +
___ COL1               CHAR               3      __ __ _____ ___________________


****************************** Bottom of data ********************************


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Cmd

Line command area. Use the following line commands to manipulate the panel rows containing column information:

I*n*

Insert *n* new rows.

R*n*

Repeat the current row *n* times. (See following note.)

D*n*

Delete *n* rows.

Note: When you repeat a row, ZDT/Db2 repeats only the column information shown on this panel and the Nulls and Default Values panel. All other information is cleared.

Column Name

The name of the column.

Data Type

The data type of the column.

Len

The length of the column being defined. Only use this field for columns with a string data type.

Pr, Sc

The precision and scale for numeric data types.

For a decimal number:

- Specify the precision (1–31) in the **Pr** field and the scale (0 to the precision of the number) in the **Sc** field.

For a single-precision floating point number:

- If you have specified a data type of FLOAT, specify the precision (1–21) in the **Pr** field and leave the **Sc** field blank.
- If you have specified a data type of REAL, leave the **Pr** and **Sc** fields blank.

For a double-precision floating point number:

- If you have specified a data type of FLOAT, specify the precision (22–53) in the **Pr** field or leave the **Pr** field blank, and leave the **Sc** field blank.
- If you have specified a data type of DOUBLE, leave the **Pr** and **Sc** fields blank.

For a binary integer, leave the **Pr** and **Sc** fields blank.

For Data

The subtype to be used in the FOR *subtype* DATA clause. Only specify a non-blank value for character string columns with a data type of CHAR, VARCHAR, LONG VARCHAR, or CLOB. Valid values are:

(Blank)

FOR DATA clause not generated.

SBCS (or S)

FOR SBCS DATA clause generated.

MIXED (or M)

FOR MIXED DATA clause generated.

BIT (or B)

FOR BIT DATA clause generated. Do not specify BIT for a CLOB column.

User-defined Data Type

Specifies the data type of the column is a distinct type (a user defined-data type). The length, precision and scale of the column (if applicable) are respectively the length, precision and scale of the source type of the distinct type.

Parent panels

- Create Table panel on page 561

Child panels

None.

Related tasks

- Creating a table on page 226

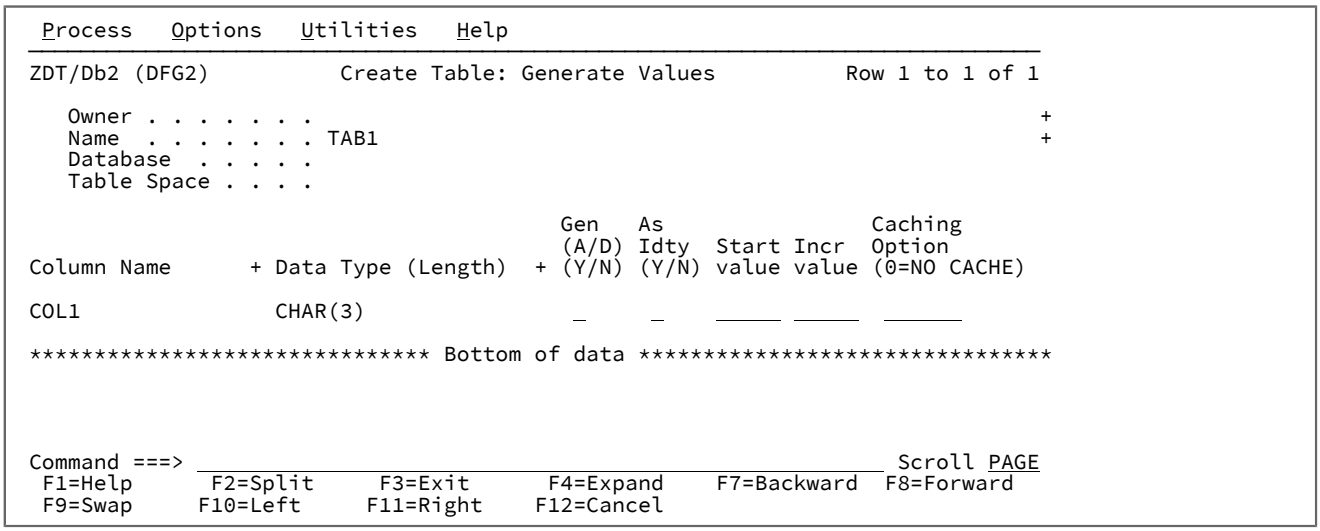
Related references

- Db2 data types on page 440
- Create Table: Nulls and Default Values panel on page 584

Create Table: Generate Values panel

You use the Create Table: Generate Values panel to specify information to define generated values for identity or ROWID columns in the table.

Panel and field definitions



Column Name

The name of the column in the table you are creating for which you want Db2® to create values.

Gen (A/D)

Determines when Db2® generates values for the column. Valid values are:

(blank)

Values not generated for this column.

A

Values always generated for this column when a new row is inserted into the table.

D

Values generated for this column when a new row is inserted unless a value is specified.

As Idty (Y/N)

Identifies whether the column is the identity column for the table (there can be only one).

N (or blank)

Column is not an identity column.

Y

Column is the identity column for the table.

Start value

Starting value for generating identity column values (integer).

Incr value

Interval between consecutive generated identity column values (non-zero integer).

Caching Option

An integer specifying how many preallocated values of the identity column are kept in memory to improve performance. Valid values:

0

Generates a NO CACHE clause.

n

Generates a NO CACHE *n*clause. *n* must be 2 or more.

Parent panels

- Create Table panel on page 561

Child panels

None.

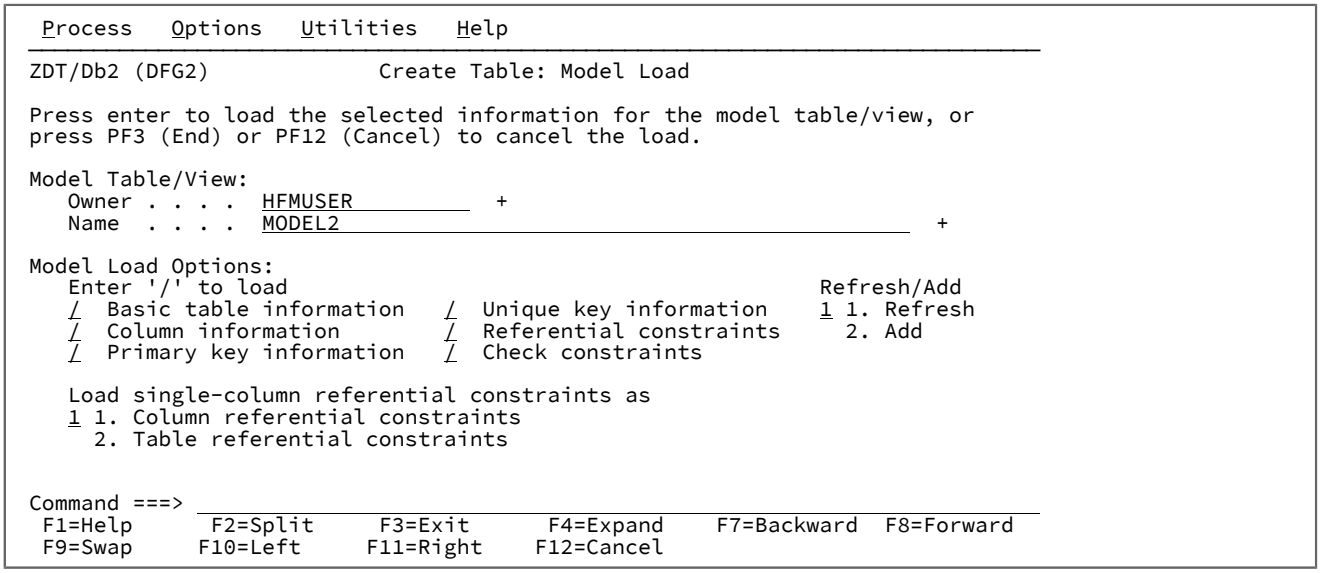
Related tasks

- Creating a table on page 226

Create Table: Model Load panel

You use the Create Table: Model Load panel to control what information about a model table is loaded into the ISPF variables and tables used to generate the CREATE TABLE statement. The panel also allows you to specify whether the load is a refresh of the current information, or adds to it.

Panel and field definitions



Owner

The name of the owner of the model table.

Name

The name of the model table.

Basic table information

Database name, table space name if explicitly defined for the model table, and all the table options values for the model table or view. (See following Note.)

Column information

Column names and data types, and nulls and default values specifications for the model table or view. (See following Note.)

Primary key information

Column order in the primary key of the model table.

Unique key information

Column order for the first three unique keys in the model table.

Referential constraints

Referential constraints for the model table.

Check constraints

Check constraints for the model table.

1. (Refresh)

Discard all current CREATE TABLE information, whether manually entered or from a previous load, and load the new selected information for the model table or view.

2. (Add)

Add the selected type of information for the model table or view to the current information for the new table.

Note: If you select the Add option, you can only select **Basic table information** and **Column information**. ZDT/Db2 only loads basic table information if the target field is blank.

Load single-column referential constraints as

If the model table has referential constraints, whether you want single-column referential constraints loaded as *column* referential constraints or *table* referential constraints.

1. Column referential constraints

Single-column referential constraints associated with the foreign key column.

2. Table referential constraints

Single-column referential constraints not associated with the foreign key column. (The associated constraint names are kept regardless of the level of the currently connected Db2® system.)

Parent panels

- Create Table panel on page 561

Child panels

None.

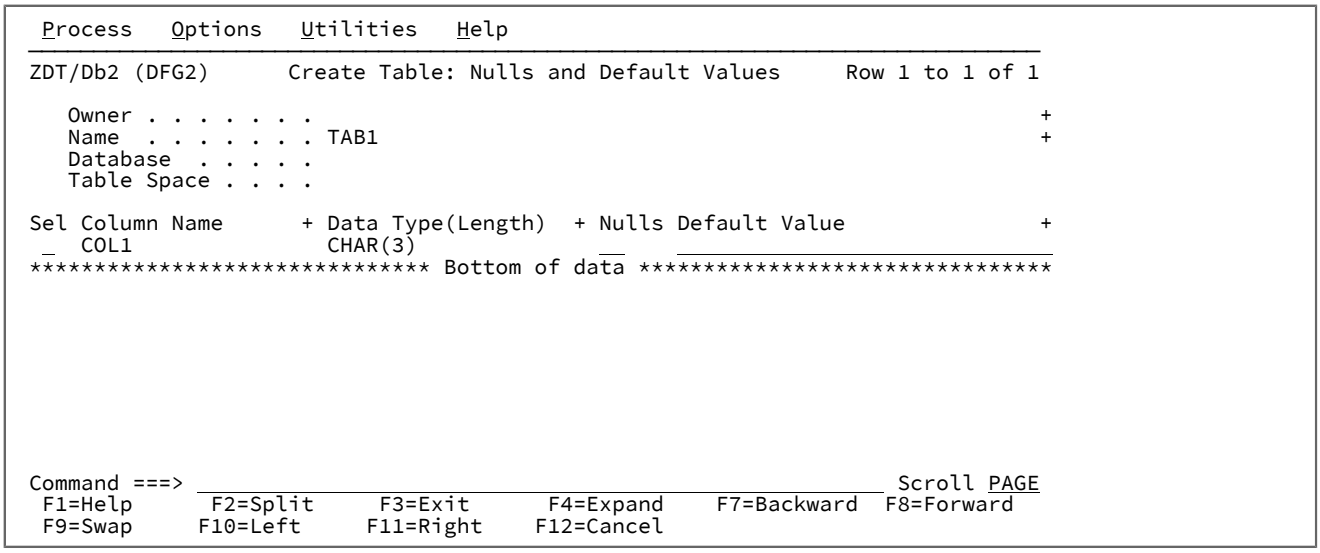
Related tasks

- Creating a table on page 226

Create Table: Nulls and Default Values panel

You use the Create Table: Nulls and Default Values panel to specify information to define the null attributes and default values for the columns of the table.

Panel and field definitions



Owner

The name of the owner of the table.

Name

The name of the table.

Database

The name of the database in which the table is to be created.

Table Space

The name of the table space in which the table is to be created.

Sel

Type `s` in this field to display a panel that you use to enter a long default value (up to 512 characters) for the column. See the description for the **Default Value** field.

Column Name

The name of the column.

Data Type (Length)

The data type and, where applicable, the length of the column.

Nulls

Determines if the field can contain null values, and the default value (if any) to be used for the column when a row is inserted in the table without explicitly specifying a value for the column. Valid values are:

(blank)

Permit nulls and use null as the default value. No NOT NULL or WITH DEFAULT clause is generated in the CREATE TABLE statement.

For non-identity columns, leaving the **Nulls** field blank and specifying DN are equivalent. For identity columns only, a blank value in the **Nulls** field is equivalent to specifying NN.

NN

Disallow nulls, with no default value. A **N**OT **N**ULL clause is generated in the CREATE TABLE statement.

ND

Disallow nulls, and use the default value in the **Default Value** field or the system-defined non-null default value. A **N**OT NULL WITH **D**EFAULT clause is generated in the CREATE TABLE statement.

NU

Disallow nulls, and use the value of the user special register at the time of insert or load as the default value. A **N**OT NULL WITH DEFAULT **U**SER clause is generated in the CREATE TABLE statement.

NS

Disallow nulls, and use the SQL authorization ID of the process at the time of insert or load as the default value. A **N**OT NULL WITH DEFAULT CURRENT **S**QLID clause is generated in the CREATE TABLE statement.

DN

Permit nulls and use null as the default value. A **D**EFAULT **N**ULL clause is generated in the CREATE TABLE statement.

DD

Permit nulls, and use the default value in the **Default Value** field or the system-defined non-null default value. A WITH **D**EFAULT clause is generated in the CREATE TABLE statement.

DU

Permit nulls, and use the value of the user special register at the time of insert or load as the default value. A WITH **D**EFAULT **U**SER clause is generated in the CREATE TABLE statement.

DS

Permit nulls, and use the SQL authorization ID of the process at the time of insert or load as the default value. A WITH **D**EFAULT CURRENT **S**QLID clause is generated in the CREATE TABLE statement.

Default Value

The value that a column contains when a row is inserted. If you specify a constant, you must specify either ND or DD in the **Nulls** field; otherwise, leave this field blank.

If you need to specify a default value that is too long to fit in the field on the panel, enter `s` in the **Sel** field to display a panel. If a long default value already exists and you alter the part of the value displayed in the **Default Value** field, ZDT/Db2 displays the same panel that you can use to confirm the changes you have made.

The following points apply to the **Default Value** field:

- In most cases, when you specify a character string as a default value, you do not need to enclose the string in quotation marks. ZDT/Db2 places quotation marks around unquoted character strings when it generates the SQL.
- You must enclose a character string in quotation marks when the string contains trailing blanks.

  Specify "Twenty characters ", as:

  ```
  'Twenty characters   '
  ```

- If the string contains quotation marks, represent each quotation mark with two quotation marks.

  Specify *"It's Diana's house"* as:

  ```
  It''s Diana''s house
  ```

  Specify *"Captain Cook sailed on the 'Endeavour'"* as:

  ```
  Captain Cook sailed on the ''Endeavour''
  ```

  Note:

If the string starts and ends with quotation marks, place the entire string within quotation marks and represent each quotation mark contained within the string with two quotation marks.

Specify *"'Harry's Practice'"* as:

```
''Harry''s Practice''
```

- If the data type of the column is a distinct type, specify the default value only (ZDT/Db2 generates the required casting function for the distinct type). If the distinct type is based on a string data type, enclose the constant within quotation marks.

Parent panels

- Create Table panel on page 561

Child panels

None.

Related tasks

- Creating a table on page 226

Create Table: Options panel

You use the Create Table: Options panel to specify information to define options for the table you are creating.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)              Create Table: Options

Processing Options, Internal Identifiers.:
EDITPROC . . .  DSN8EAE1          (optional)
VALIDPROC  . .  _________          (optional)
OBID value . .  ______             (optional)

Audit and Data Storage Options:
    Audit Options                 Data Storage Options
    _ 1. None (default)           _ 1. EBCDIC  (default)
      2. Changes                    2. ASCII
      3. All                        3. UNICODE

Logging and Table Deletion Options:
    Enter '/' to select option
    _  Log data capture changes (default is not to log)
    _  Restrict drop of table   (default is to allow)


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

EDITPROC

If the table you are creating requires an EDITPROC clause, specify the name of the edit procedure here. The default is not to use an EDITPROC clause. An edit procedure transforms data at row level within Db2® before presentation.

VALIDPROC

If the table you are creating requires a VALIDPROC clause, specify the name of the validation procedure here. The default is not to use a VALIDPROC clause. A validation procedure validates data at row level before it is added.

OBID value

Use this field, if necessary, to specify the integer for the OBID clause for the table you are creating. Db2® assigns an (internal) object identifier value to each Db2® object. There are circumstances where it is desirable to specify the value for a Db2® object instead of using the Db2-assigned value. An example is where there are two parallel Db2® subsystems, with the same objects defined in each, and where every matching object has the same OBID value.

Audit Options

Db2® provides 3 auditing options which determine the type of access to the table that causes auditing to be performed:

1. **None**

   No auditing performed. This is the default setting.

2. **Changes**

   Auditing performed when changes are made to the table.

3. **All**

   Auditing performed when the table is accessed.

Data Storage Options

The encoding scheme for string data stored in the table. ZDT/Db2 uses the selected option when it generates the CCSID clause in the CREATE TABLE statement.

1. **EBCDIC**

   String data encoded using EBCDIC CCSIDs. This is the default setting.

2. **ASCII**

   String data encoded using ASCII CCSIDs.

3. **UNICODE**

   UNICODE string data encoded using UNICODE CCSIDs.

Log data capture changes

Whether additional information is logged for INSERT, UPDATE and DELETE operations.

Slash (/)

ZDT/Db2 generates a DATA CAPTURE CHANGES clause in the CREATE TABLE statement.

(blank)

ZDT/Db2 generates a DATA CAPTURE NONE clause in the CREATE TABLE statement. This is the default setting.

Restrict drop of table

Whether the table cannot be dropped. You can use this option to prevent the accidental deletion of tables.

Slash (/)

ZDT/Db2 generates a WITH RESTRICT ON DROP in the CREATE TABLE statement.

(blank)

ZDT/Db2 allows the table to be dropped. This is the default setting.

Parent panels

- Create Table panel on page 561

Child panels

None.

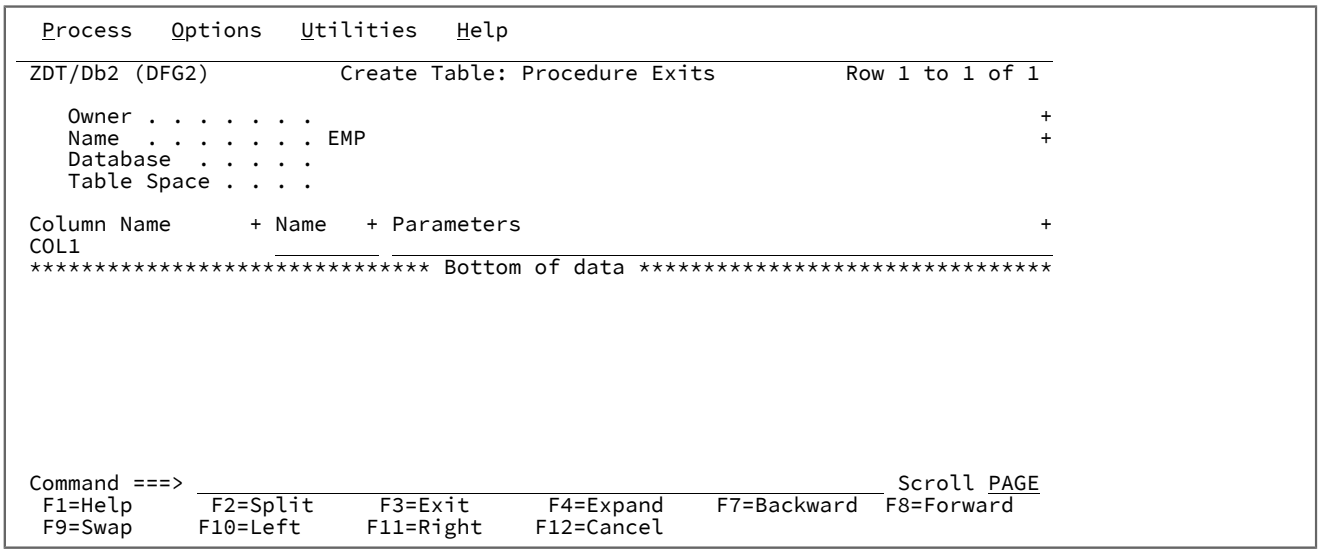
Related tasks

- Creating a table on page 226

Create Table: Procedure Exits panel

You use the Create Table: Procedure Exits panel to specify information to define procedure exits for the table you are creating.

Panel and field definitions



Column Name

The name of the column in the table you are creating.

Name

The program name of the field procedure exit.

Parameters

A list of parameters passed to the field procedure exit when it is called. The list must be comma-delimited and enclosed in parenthesis as required by the SQL syntax.

Parent panels

- Create Table panel on page 561

Child panels

None.

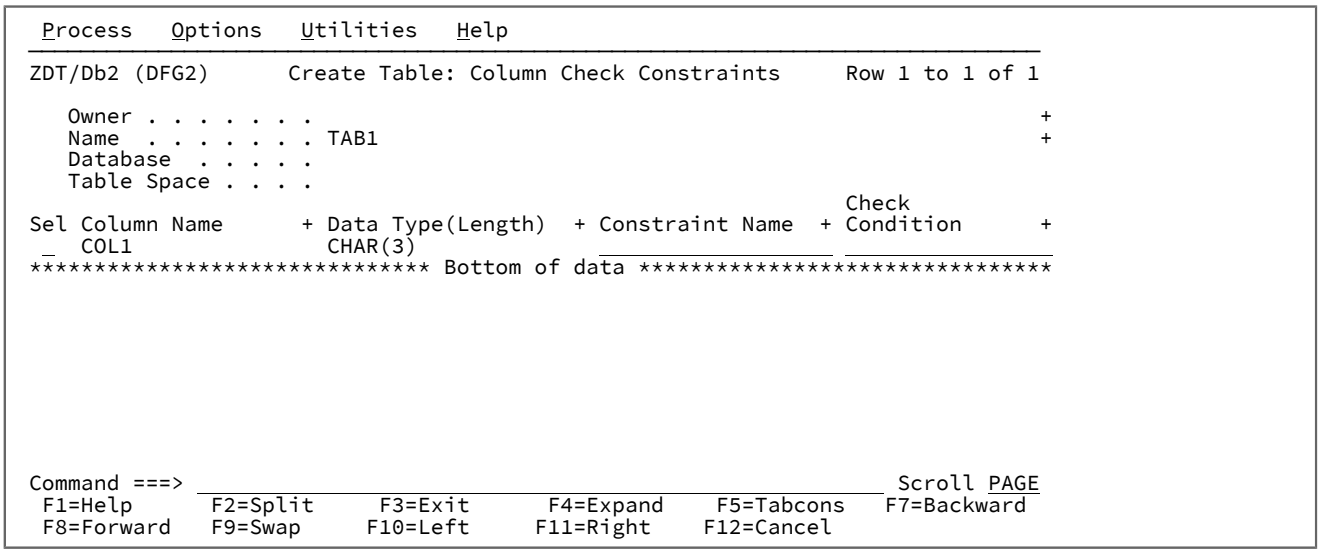
Related tasks

- Creating a table on page 226

Create Table: Table Check Constraints panel

You use the Create Table: Table Check Constraints panel to specify information to define table check constraints for the table.

Panel and field definitions



Cmd

Line command area. Use the following line commands to manipulate the panel rows containing table check constraint information:

I*n*

Insert *n* new rows.

R*n*

Repeat the current row *n* times.

D*n*

Delete *n* rows.

S

Select a row. ZDT/Db2 displays a scrollable panel that you use to specify a check condition up to the maximum length allowed by Db2®. See the description for the **Check Condition** field.

Constraint Name

The name of the constraint. This is an optional field.

Check Condition

SQL expression defining the table check constraint.

If you need to specify a check condition that is too long to fit in the field on the panel, enter `s` in the **Cmd** field to display a panel. If a long check condition already exists and you alter the part of the check condition displayed on the Table Check Constraints panel, ZDT/Db2 displays the same panel that you can use to confirm the changes you have made.

Note: ZDT/Db2 does not validate the check condition you specify.

Parent panels

- Create Table panel on page 561

Child panels

None.

Related tasks

- Creating a table on page 226

Create Table: Table Referential Constraint panel

You use the Create Table: Table Referential Constraint panel to specify the foreign key and parent key columns for a table referential constraint.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)    Create Table: Table Referential Constraint   Row 1 to 1 of 1

Constraint:
   Name  . . . . . . . . . CON1
   Parent Table Owner  . .
   Parent Table Name . . . EMP

Order Column Name        Data Type(Length) Parent Key Column
___  COL1               CHAR(3)
****************************** Bottom of data ********************************









Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Name

The name of the constraint.

Parent Table Owner

The name of the parent table owner.

Parent Table Name

The name of the parent table.

Order

Enter s or a number to add a column to the foreign key. When you enter s, ZDT/Db2 generates the next highest number for the key. To deselect a column that is already selected, clear the displayed number by overtyping it with spaces.

Note: The actual numbers are not important; ZDT/Db2 uses their relative order to define the column order for the key.

Column Name

The name of the column in the table you are creating.

Data Type(Length)

The data type of the column in the table you are creating.

Parent Key Column

Specify the name of the column in the parent table that you want to relate to the foreign key column (**Column Name**) you have ordered. If you leave this field blank, ZDT/Db2 relates the foreign key to the primary key of the parent table.

Note: If you have selected more than one column for the foreign key (by entering S or a number in the **Order** field), the corresponding Parent Key Columns must all be blank or all be filled.

Parent panels

- Create Table: Table Referential Constraints panel on page 593

Child panels

None.

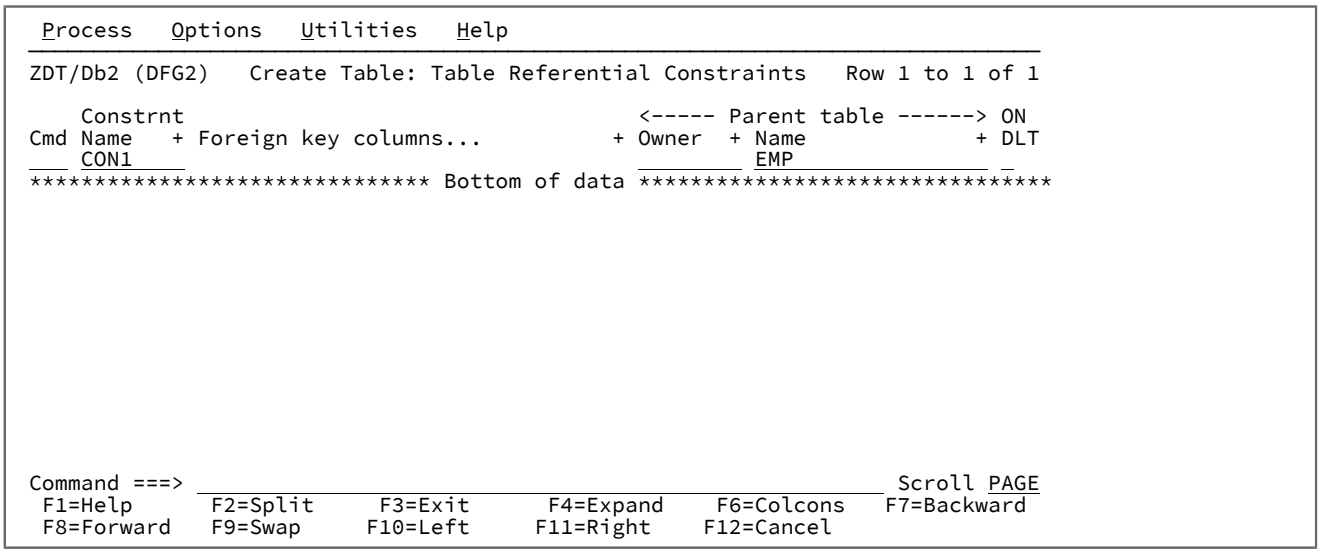
Related tasks

- Creating a table on page 226

Create Table: Table Referential Constraints panel

You use the Create Table: Table Referential Constraints panel to specify information to define one or more table referential constraints for the table.

Panel and field definitions



Cmd

Line command area. Use the following line commands to manipulate the panel rows containing table referential constraint information:

I*n*

Insert *n* new rows.

R*n*

Repeat the current row *n* times.

D*n*

Delete *n* rows.

S

Select the row and display the Referential Constraint panel.

Constrnt Name

The name of the constraint. This is an optional field.

Foreign key columns...

A list of the columns currently selected as the foreign key for this constraint. ZDT/Db2 displays only as many columns as fit in the available space.

Parent Table Owner

Optionally use this field to qualify the name of the parent table.

Parent Table Name

The name of the table containing the parent key. This field is mandatory.

ON DLT

See details for the **ON DLT** field as described in Create Table: Column Referential Constraints panel on page 576.

Parent panels

- Create Table panel on page 561

Child panels

To display this panel...	Use/do this
Create Table: Table Referential Constraint panel on page 592	Enter S in **Cmd** field

Related tasks

- Creating a table on page 226

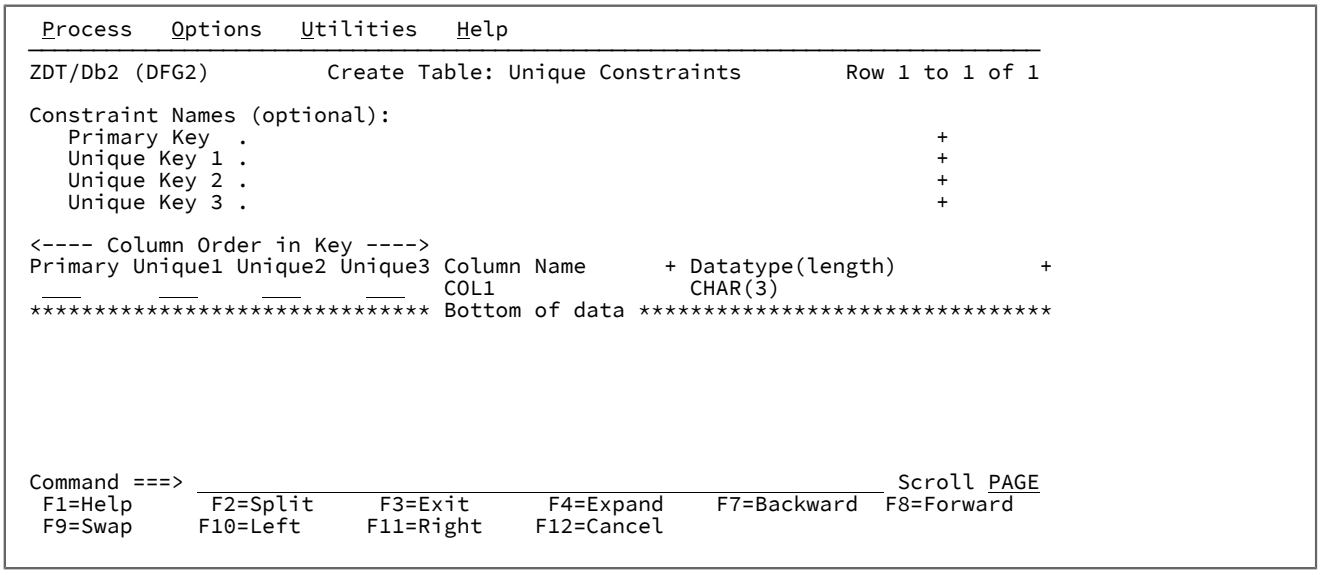
Related references

- Create Table: Table Referential Constraint panel on page 592

Create Table: Unique Constraints panel

You use the Create Table: Unique Constraints panel to specify information to define a primary key and up to three additional unique keys for the table you are creating.

Panel and field definitions



Primary

If not blank, shows the relative order of the column for the primary key.

Unique1

If not blank, shows the relative order of the column for the first unique key.

Unique2

If not blank, shows the relative order of the column for the second unique key.

Unique3

If not blank, shows the relative order of the column for the third unique key.

Column Name

The name of the column.

Datatype(length)

The data type and, if applicable, the length of the column.

Parent panels

- Create Table panel on page 561

Child panels

None.

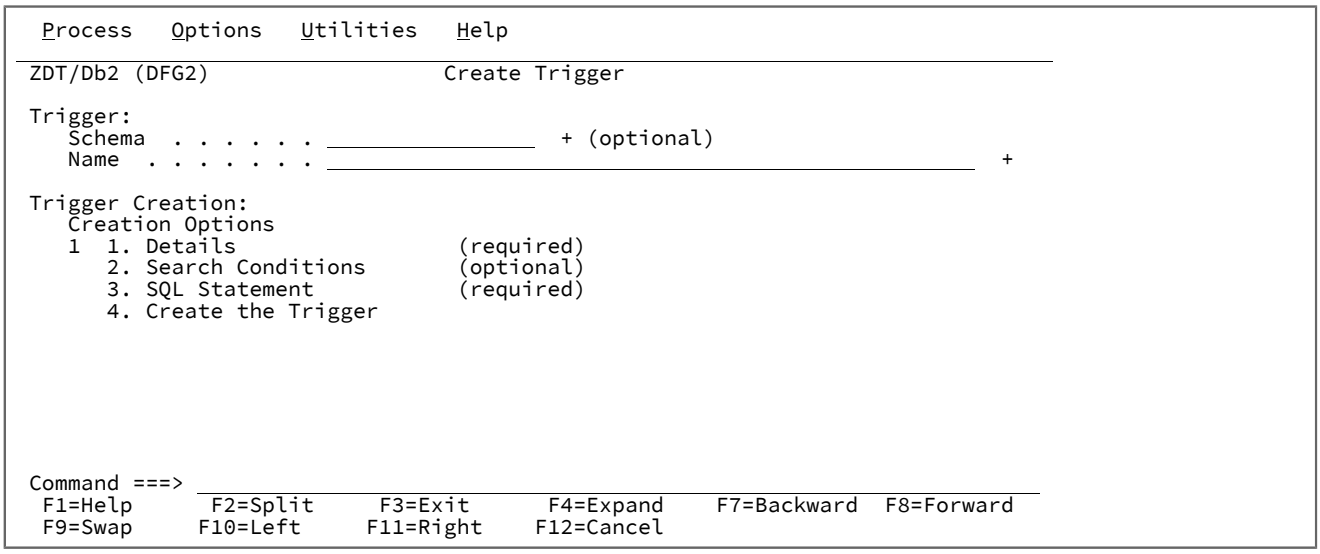
Related tasks

- Creating a table on page 226

Create Trigger panel

You use the Create Trigger panel to create a new trigger.

Panel and field definitions



Schema

Name of schema.

Name

Name of new trigger.

Creation Options

Specify the number of the creation option you want to use:

1. Details

To specify information to define the type of trigger to be created. Displays the Create Trigger: Details panel.

2. Search Conditions

To specify information to define the optional search condition to restrict the execution of the trigger. Displays the Create Trigger: Search Condition panel.

3. SQL statement

To specify SQL statements to be executed when the trigger is activated. displays the Create Trigger: SQL statement panel.

4. Create the Trigger

When you have finished specifying details for the trigger to be created, select this creation option to create the trigger.

Parent panels

- Db2 Object Functions panel on page 622
- Triggers panel on page 903

Child panels

To display this panel...	Use/do this
Create Trigger: Details panel on page 598	Creation option 1
Create Trigger: Search Condition panel on page 600	Creation option 2
Create Trigger: SQL statement panel on page 600	Creation option 3

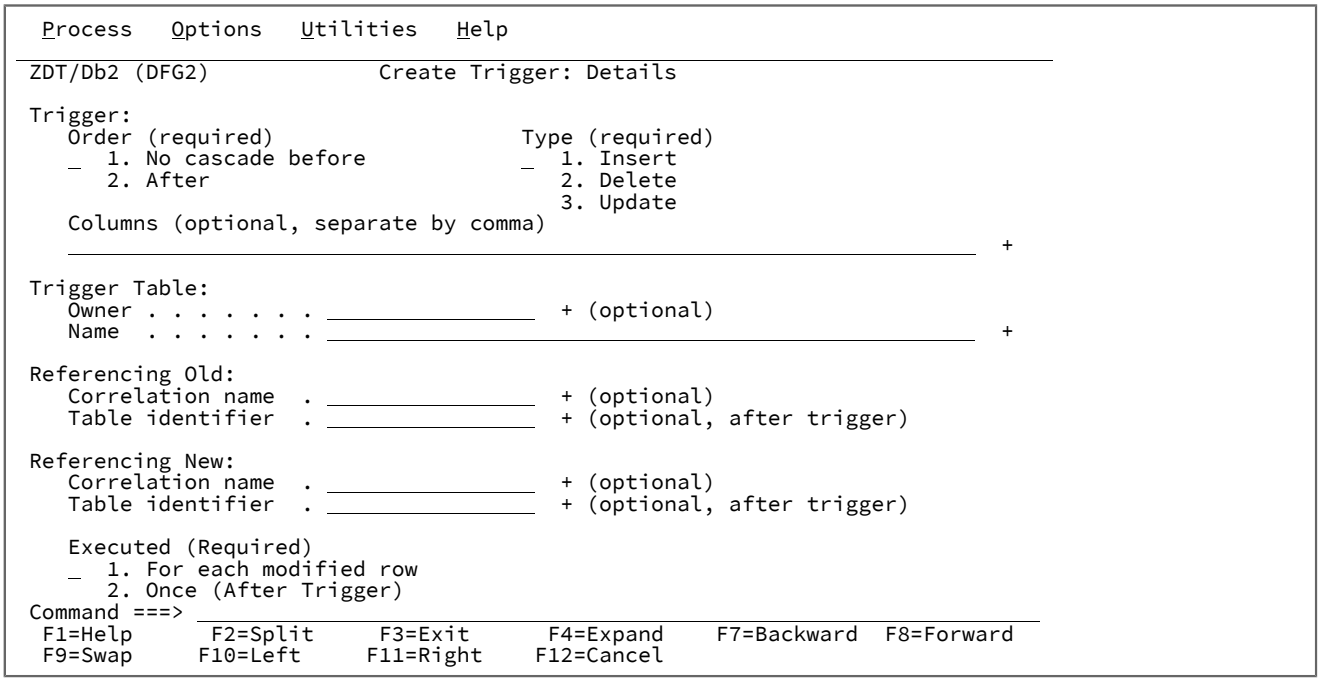
Related tasks

- Creating a trigger on page 241

Create Trigger: Details panel

You use the Create Trigger: Details panel to specify information to define the type of trigger to be created.

Panel and field definitions



No cascade before

Defines the trigger as a before trigger.

After

Defines the trigger as an after trigger.

Type

Insert

The trigger is executed when a row is inserted in the trigger table.

Delete

The trigger is executed when a row is deleted from the trigger table.

Update

The trigger is executed when a row is updated from the trigger table.

Columns

Specifies a list of columns to restrict the execution of an update trigger to updates on the specified columns.

Name

Name of the table against which an action causes the trigger to be executed.

Owner

Owner of the trigger table.

Old correlation name

Correlation name used to reference rows as they were before the triggering SQL operation.

New correlation name

Correlation name used to reference rows as they were after the triggering SQL operation.

Old table identifier

Temporary table identifier used to reference the complete set of rows as they were before the SQL operation.

New table identifier

Temporary table identifier used to reference the complete set of rows as they were after the SQL operation.

1. For each modified row

The trigger option is executed for each modified row of the trigger table.

2. Once (After Trigger)

The trigger action is executed only once for the triggering SQL operation.

Parent panels

- Create Trigger panel on page 596

Child panels

None.

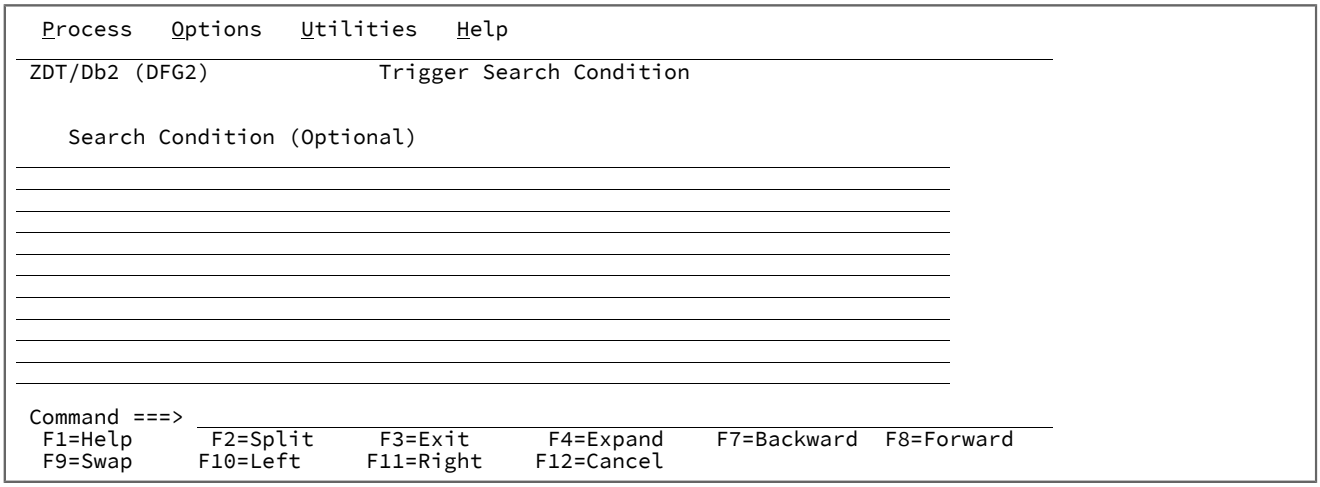
Related tasks

- Creating a trigger on page 241

Create Trigger: Search Condition panel

You use the Create Trigger: Search Condition panel to specify information to define the optional search condition to restrict the execution of the trigger. You do not need to type the WHEN keyword and enclosing parentheses for the condition; they are added automatically if omitted.

Panel and field definitions



Search Condition

Freeform entry area for the search condition. The triggered SQL statements are executed only if the search condition evaluates to true or is omitted.

Parent panels

- Create Trigger panel on page 596

Child panels

None.

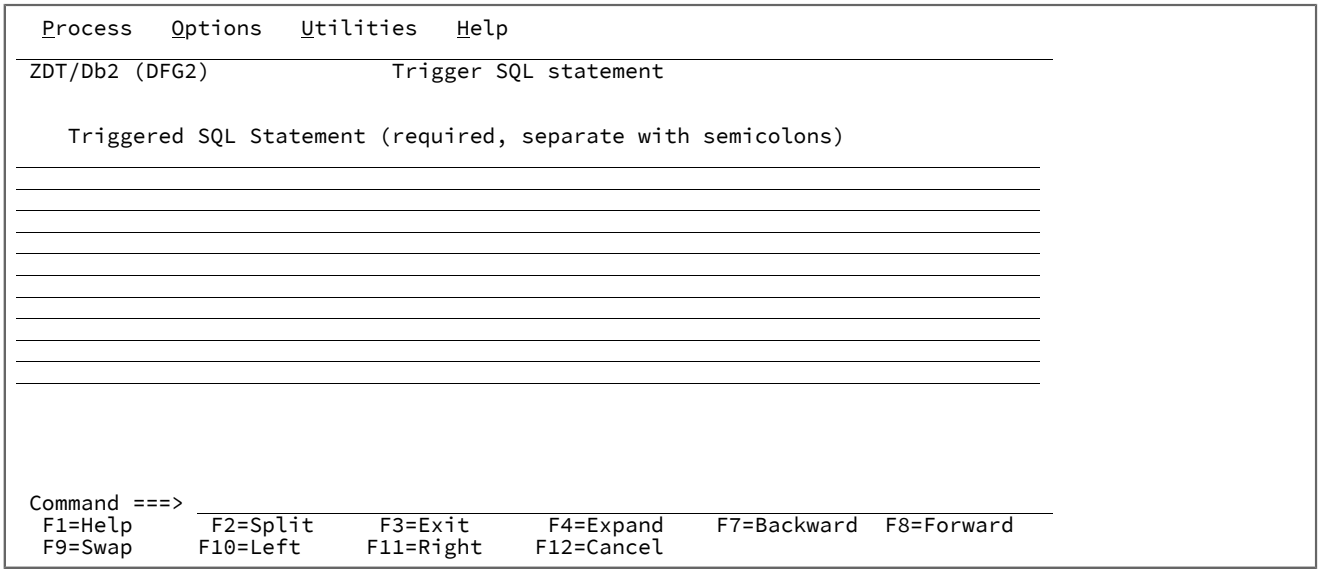
Related tasks

- Creating a trigger on page 241

Create Trigger: SQL statement panel

You use the Create Trigger: SQL statement panel to specify SQL statement(s) to be executed when the trigger is activated. You must use a semicolon to separate multiple SQL statements, but you do not need to enclose the statements in BEGIN ATOMIC … END keywords; they are added automatically if omitted.

Panel and field definitions



Triggered SQL Statements

Freeform entry area for the trigger SQL. Enter one or more SQL statements separated by semicolons.

Parent panels

- Create Trigger panel on page 596

Child panels

None.

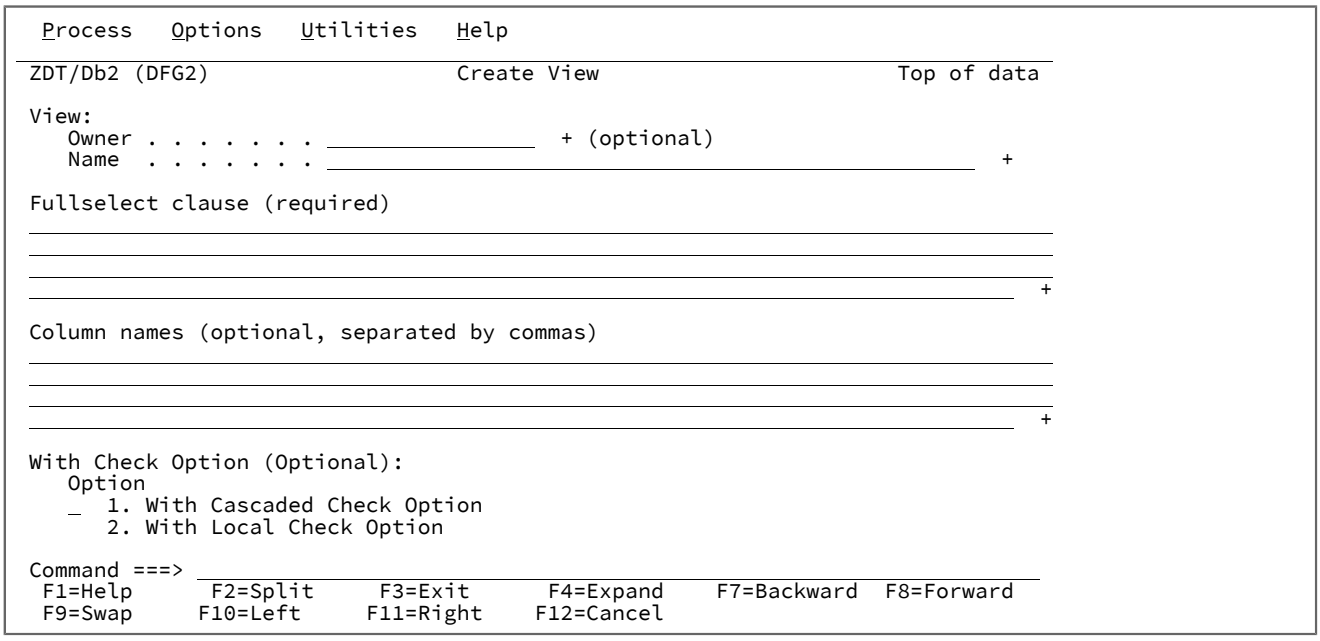
Related tasks

- Creating a trigger on page 241

Create View panel

You use the Create View panel to create a new view.

Panel and field definitions



Owner

The name of the owner of the view.

Name

The name of the view to be created.

Subselect clause

SQL subselect clause defines the contents of the view.

Column names

A list of column names defining the names of the view columns. Must be specified if the subselect clause result table contains duplicate or unnamed columns. If specified, must contain the same number of columns as the subselect clause result table.

With Check Option

Controls whether or not inserted rows must conform to the view definition.

1. **With Cascaded Check Option**

   Generates WITH CASCADED CHECK OPTION clause.

2. **With Local Check Option**

   Generates WITH LOCAL CHECK OPTION clause.

Parent panels

- Db2 Object Functions panel on page 622

Child panels

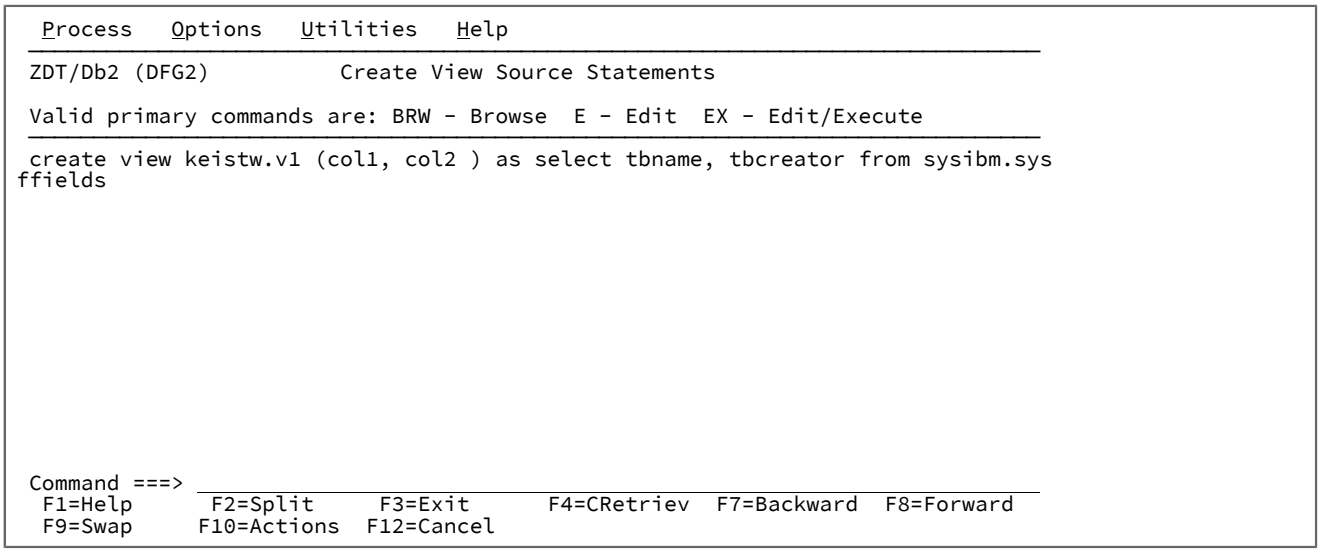
None.

Related tasks

- Creating a view on page 235

Create View Source Statements panel

Panel and field definitions



Parent panels

- Tables, Views and Aliases panel on page 891

Child panels

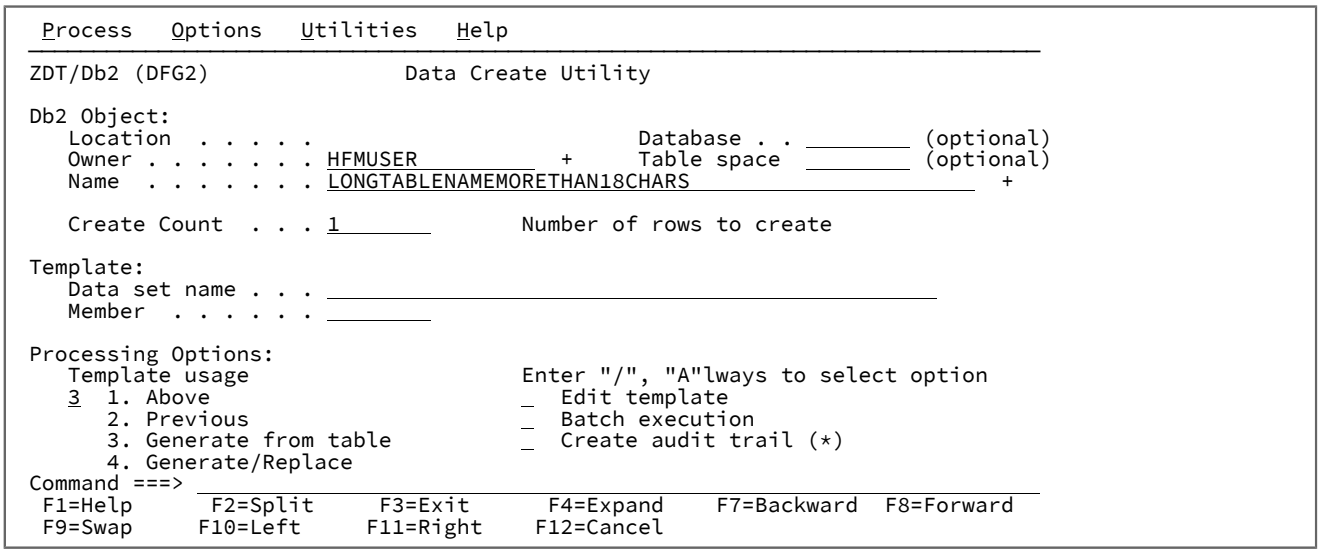
None.

Related tasks

Data Create Utility panel

You use the Data Create Utility panel to populate a Db2® table with a specified number of rows of data.

Panel and field definitions



Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Create Count

Specify the number of rows you want to create in the Db2® table. This is a required field and must be in the range 1–2147483647. The default is 1.

Data set name

If you are using a specific template to specify the data initialization details, (see option 1 in **Template usage**), specify the name of the template data set. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Processing Options

You can use these options to specify which template you want to use for the create process, and whether you want to edit the template before the create process.

Template usage

The Data Create Utility panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. Previous

Uses the last (previously used) template for this table.

3. Generate from table

Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **Db2® Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a / or an "A":

Edit template

Use to edit the template before creating the data. The default is not selected.

Batch execution

Builds the necessary JCL to perform the Create function in a batch job. The generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing. The default is not selected.

Create audit trail

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set

the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Parent panels

- Utility Functions panel on page 915

Child panels

To display this panel…	Use/do this
Column Selection/Edit panel on page 497	Select **Edit template**

Related tasks

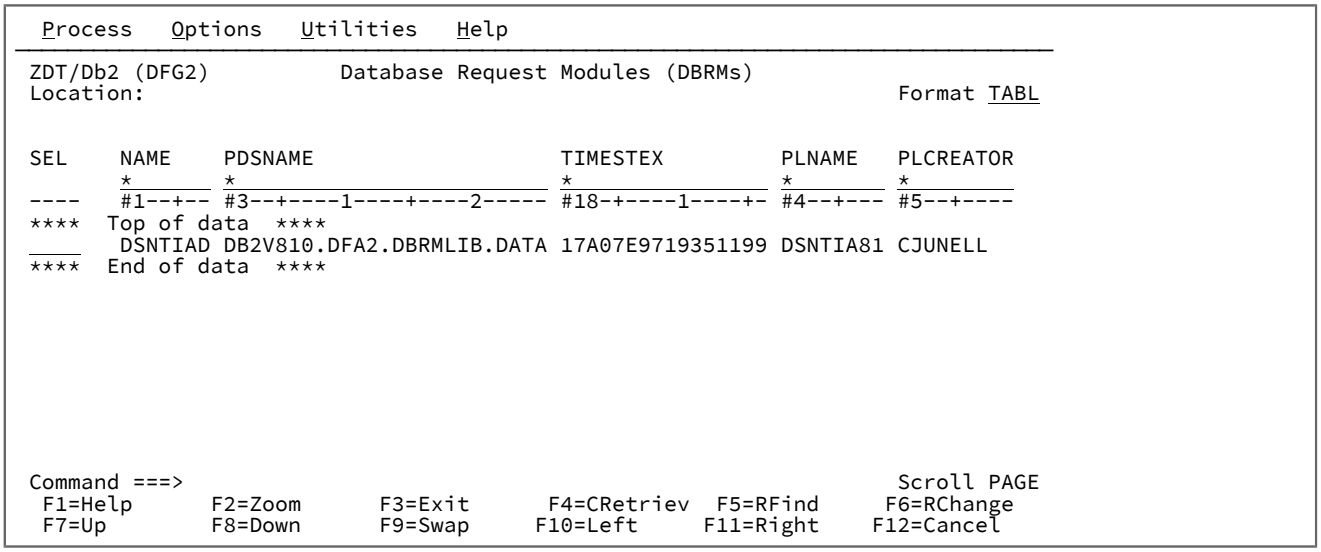
- Populating a Db2 table with data on page 244
- Specifying a data set and a member name on page 52
- Selecting options on ZDT/Db2 panels on page 45
- Working with templates on page 74
- SAF-rule controlled auditing on page 71

Database Request Modules panel

You use the Database Request Modules panel to list the DBRMs in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSDBRM.



Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Application Plans panel on page 473

Child panels

To display this panel...	Use/do this
(ISPF panel)	Line command B
Details panels on page 636	Line command I
Application Plans panel on page 473	Line command PL
Display Row panel on page 637	Line command ROW
	Primary command SORT

To display this panel...	Use/do this
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Database Structure panel

ZDT/Db2 displays the Database Structure panel if you use the DS line command against an item listed on:

- The Databases panel on page 609, or
- The Table Spaces panel on page 879

Databases panel

You use the Databases panel to list database object types in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSDATABASE.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Databases
Location:                                                    Format TABL

                                                                      GROUP+
SEL    NAME     CREATOR  STGROUP     BPOOL      DBID CREATEDBY TYPEEX   MEMBER
       *        *        *           *           *   *         *        *
----   #1--+--- #2--+--- #3--+----1- #4--+--- ----#5 #7--+---- #20-+--- #11-+-
****  Top of data  ****
_____  ABC      SHRIKE2  DDDDDDDDDDD BP0         270 SHRIKE2
_____  DSN8D81A CJUNELL  DSN8G810    BP0         259 CJUNELL
_____  DSN8D81P CJUNELL  DSN8G810    BP0         260 CJUNELL
_____  DSNATPDB CJUNELL  SYSDEFLT    BP0         257 CJUNELL
_____  DSNDB04  SYSIBM   SYSDEFLT    BP0           4 SYSIBM
_____  DSNDB06  SYSIBM                           6 SYSIBM
_____  DSNDB07  CJUNELL  SYSDEFLT    BP0           7 CJUNELL   WORKFILE
_____  DSNRGFDB CJUNELL  SYSDEFLT    BP0         258 CJUNELL
_____  DSNRLST  CJUNELL  SYSDEFLT    BP0         256 CJUNELL
_____  FUNTBDB  SHRIKES  SYSDEFLT    BP8K0       271 SHRIKES
_____  JLV3D81A SHRIKES  JLV3G810    BP0         266 SHRIKES
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Table Spaces panel on page 879
- Tables, Views and Aliases panel on page 891
- Indexes panel on page 723

Child panels

To display this panel…	Use/do this
Alter Database panel on page 453	Line command A
Create Database panel on page 529	Line command CR
Create Table Space panel on page 564	Line command CRS
See Drop panels on page 640	Line command DR

To display this panel…	Use/do this
Database Structure panel on page 609	Line command DS
Grant privileges panels on page 715	Line command G
Generate SQL From Db2 Catalog panel on page 712	Line command GEN
Details panels on page 636	Line command I
Recovery Information panel on page 787	Line command ICS
Privileges panels on page 768	Line command P
Revoke privileges panels on page 798	Line command R
Display Row panel on page 637	Line command ROW
Table Spaces panel on page 879	Line command S
Tables, Views and Aliases panel on page 891	Line command T
Indexes panel on page 723	Line command X

To display this panel...	Use/do this
	Primary command SORT

Sort Fields panel on page 841

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Db2® Browse panel

You use the Db2® Browse panel to enter the name of an object to be browsed. You also use this panel to specify the name of a template that describes a logical view of the object.

Panel and field definitions

```
 Process   Options   Utilities   Help
-------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Db2 Browse

Specify the Db2 Object:
   Location  . . . . . _______________          Database . . ________ (optional)
   Owner . . . . . . . HFMUSER          +       Table space  ________ (optional)
   Name  . . . . . . . EMP                                                    +
   Start position  . . 1
   Row count . . . . . 100          Number of rows to browse

Template:
   Data set name . . . 'HFMUSER.TEMPLATE'
   Member  . . . . . . SALARY

Processing Options:
   Template usage                   Enter "/", "A"lways to select option
   3  1. Above                      _  Edit options
      2. Previous                   _  Edit template
      3. Generate from table        _  Re-edit template
      4. Generate/Replace
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Start position

Indicates where ZDT/Db2 is to begin loading rows from the result table into the ZDT/Db2 editor.

If you specify a value greater than 1 (the default), ZDT/Db2 skips n-1 rows of the result table before loading rows into the ZDT/Db2 editor. If the **Row count** value is `ALL` (or `0` or `*`), when the data is displayed you can scroll up to see the rows that were skipped. For a fixed **Row count** value, you cannot scroll backwards to see any skipped rows.

Row count

Use this option to specify the maximum number of rows to be retrieved from Db2® and loaded into memory for the Browse session. You can specify a value in the range 1–999999999 or, to indicate that all rows are to be retrieved, specify `ALL`, `0`, or `*`. The default value is 100.

A Browse session only applies to those rows fetched from Db2® and loaded into the ZDT/Db2 editor. It does not apply to any unfetched rows. This means that the FIND primary command only applies to those rows fetched from Db2®.

Note: When you are browsing large tables, use the value of `ALL`, `0`, or `*` with caution as it is possible to encounter storage problems. The amount of storage required by the editor depends on both the number of rows retrieved, and the row length. Use the row selection criteria section of the template to limit the number of rows retrieved.

Data set name

The name of a data set that contains, or will contain, the template. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The Db2® Browse panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select this option to change the options for the current editor session:

Edit options

Enter one of these:

A / or an "A"

Displays the first ZDT/Db2 editor options panel, prior to displaying the data for the Db2® object being processed.

A number in the range 1–8

Displays the *n*th ZDT/Db2 editor options panel directly where *n* is the value entered.

You can select the following options by entering a / or an "A":

Edit template

Select this option to edit the template before browsing the data.

Re-edit template

Select this option to control the navigation between panels within the Browse dialog. This option controls the behavior of the F3 and F12 keys when pressed from within a Browse session.

Parent panels

- Table Browse panel on page 866 (when BROWSE primary command entered)
- Table Edit panel on page 868 (when BROWSE primary command entered)
- Table View panel on page 882 (when BROWSE primary command entered)
- Primary Option Menu panel on page 760 (when "B" entered on command line)
- Tables, Views and Aliases panel on page 891 (when B line command entered)

Child panels

To display this panel…	Use/do this
Table Browse panel on page 866	No options selected and press Enter
Editor Options (1 of 8) panel on page 644	Select **Edit options**
Column Selection/Edit panel on page 497	Select **Edit template** or **Re-edit template**

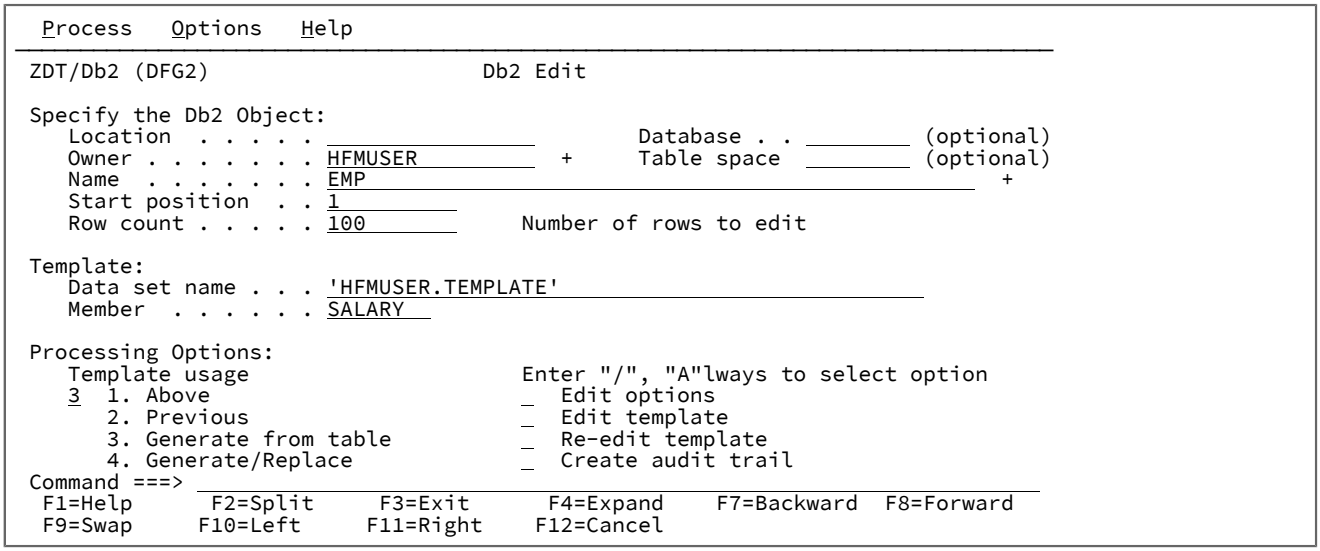
Related tasks

- Viewing and changing Db2 data on page 122
- Selecting options on ZDT/Db2 panels on page 45
- Specifying a data set and a member name on page 52
- Where you can use templates on page 22
- Working with templates on page 74
- View and Edit options (options 1 and 2) on page 67

Db2® Edit panel

You use the Db2® Edit panel to enter the name of an object to be edited. You also use this panel to specify the name of a template that describes a logical view of the object.

Panel and field definitions



Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Start position

Indicates where ZDT/Db2 is to begin loading rows from the result table into the ZDT/Db2 editor.

If you specify a value greater than 1 (the default), ZDT/Db2 skips n-1 rows of the result table before loading rows into the ZDT/Db2 editor. If the **Row count** value is `ALL` (or `0` or `*`), when the data is displayed you can scroll up to see the rows that were skipped. For a fixed **Row count** value, you cannot scroll backwards to see any skipped rows.

Row count

Use this option to specify the maximum number of rows to be retrieved from Db2® and loaded into memory for the editor session. You can specify a value in the range 1–999999999 or, to indicate that all rows are to be retrieved, specify `ALL`, `0`, or `*`. The default value is 100.

A Z Data Tools/Db2 editor session only applies to those rows fetched from Db2® and loaded into the editor. It does not apply to any unfetched rows. This means that the FIND and CHANGE primary commands only apply to those rows fetched from Db2®.

Note: When you are editing large tables, use the value of 0 with caution as it is possible to encounter storage problems. The amount of storage required by the editor depends on both the number of rows retrieved, and the row length. Use the row selection criteria section of the template to limit the number of rows retrieved.

Data set name

The name of a data set that contains, or will contain, the template. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member vname or a member name pattern.

Template usage

The Db2® Edit panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

Edit options

The default is not selected. Select this option to change the options for the current editor session by entering one of the following values:

A / or an "A"

Displays the first ZDT/Db2 editor options panel, prior to displaying the data for the Db2® object being processed.

A number in the range 1–8

Displays the *n*th ZDT/Db2 editor options panel directly where *n* is the value entered.

Any options that you set last only for the duration of the current Edit session, and override any conflicting global options. To permanently change the Edit options, select option 0.2.

You can select the following options by entering a / or an "A":

Edit template

Select this option to edit the template before editing the data. The default is not selected.

Re-edit template

Select this option to control the navigation between panels within the Edit dialog. This option controls the behavior of the F3 and F12 keys when pressed from within an Edit session. The default is not selected.

Create audit trail



Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Determines if ZDT/Db2 generates an audit trail report of all successful modifications to Db2® data made during an Edit session:

/

Audit trail report generated.

(blank)

Audit trail report not generated.

Note:
The ability to change this option depends on installation options within the Db2® subsystem to which ZDT/Db2 is connected. The **Create an audit trail** option can be:

- Permanently selected (*"/"* shown in the option). Audit trail reporting occurs for Edit processing. You cannot change the setting of the option.
- Permanently deselected (blank shown in the option). Audit trail reporting does not occur for Edit processing. You cannot change the setting of the option.
- Optional. Type a *"/"* in the option to generate audit reporting for all Edit processing; otherwise, type a space in the option to suppress audit trail reporting of any Edit processing.

At the end of your ZDT/Db2 session, the **Create an audit trail** option that you specify can be saved to your ISPF profile, or it can be set to blank (audit trail report not selected), depending on the installation options within the Db2® subsystem to which ZDT/Db2 is connected. The audit installation options for the currently connected Db2® system can be determined by selecting option 7 from the HELP pull down menu. When the auditing option is *"OPTIONAL,OFF"*, the Create an audit trail option is always reset to blank (not selected) at the start of a Z Data Tools/Db2 session. This means that, for an audit log to be written, auditing must be explicitly selected each time a new ZDT/Db2 session is started. Further, when this option is selected, there is no distinction between the global setting for the audit option, and any value entered using 'edit options' on the Db2® edit panel. For more information, see the *Z Data Tools Customization Guide* or see your systems administrator.

Parent panels

- Primary Option Menu panel on page 760
- Tables, Views and Aliases panel on page 891

Child panels

To display this panel…	Use/do this
Table Edit panel on page 868	No options selected and press Enter
Editor Options (1 of 8) panel on page 644	Select **Edit options** by typing a /, an "A", or "1"
Editor Options (2 of 8) panel on page 649	Select **Edit options** by typing a "2"
Editor Options (3 of 8) panel on page 655	Select **Edit options** by typing a "3"
Editor Options (4 of 8) panel on page 662	Select **Edit options** by typing a "4"

To display this panel...	Use/do this
Editor Options (5 of 8) panel on page 667	Select **Edit options** by typing a “5”
Editor Options (6 of 8) panel on page 672	Select **Edit options** by typing a “6”
Editor Options (7 of 8) panel on page 675	Select **Edit options** by typing a “7”
Column Selection/Edit panel on page 497	Select **Edit template** or **Re-edit template**

Related tasks

- Viewing and changing Db2 data on page 122
- Selecting options on ZDT/Db2 panels on page 45
- Specifying a data set and a member name on page 52
- Where you can use templates on page 22
- Working with templates on page 74
- View and Edit options (options 1 and 2) on page 67

Db2® Location Selection panel

You use the Db2® Location Selection panel to display all the available remote locations. You can only access remote locations that have been defined to your local Db2® subsystem.

Panel and field definitions

```
 Process   Opti ┌──────────────────── Db2 Location Selection ────────────────────┐
─────────────── │                                                    Row 1 to 4 of 4
ZDT/Db2 (DFG2)  │ Select one of the Db2 locations from the list.
                │
Specify the Db2 │ Sel Location         Linkname LU Name  IP Address
   Location  .  │  —  QXPMVS6DB2D      MVS6DB2D ?        ?
   Owner . . .  │  —  QXPMVS8DB2D      MVS8DB2D MVS8DB2D ?
   Name  . . .  │  —  QXPMVS8DB26      MVS8DB26 ?        ?
                │  —  QXPMVS8DB62      MVS8DB62 MVS8DB62 ?
Template:       │********************* Bottom of data **********************
   Data set nam │
   Member  . .  │
                │
Processing Opti │
   Template usa │
   1  1. Above │
      2. Previo │
      3. Genera │
      4. Genera │
                │
                │ Command ===> ____________________________________________________
Command ===>    │  F1=Help        F2=Split       F3=Exit        F7=Backward
 F1=Help      F │  F8=Forward     F9=Swap       F12=Cancel
 F9=Swap     F1 └──────────────────────────────────────────────────────────────────┘
```

Parent panels

Asterisk (*) or pattern entered in **Location** field on any ZDT/Db2 panel (except) Object List Utility.

Child panels

None.

Related tasks

- Specifying a Db2 object name on page 46
- Remote Db2 Location Selection panel on page 791

Db2® Object Functions panel

You use the Db2® Object Functions panel to create and drop Db2® objects.

Panel and field definitions

```
  Process   Options   Utilities   Help
 ZDT/Db2 (DFG2)              Db2 Object Functions

        C Create object                      D Drop object

 Processing Options:
    Object Type
    __   1. Database         5.  Alias            9.  Function
         2. Table space      6.  Index            10. Stored procedure
         3. Table            7.  Synonym          11. Trigger
         4. View             8.  Distinct type    12. Auxiliary Table (Create)


    Enter "/" to select option
    _  Confirm object drop




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

You can select the following option by entering a "/":

Object Type

Used to specify the type of Db2® object to process. The default is 3 (**Table**).

Confirm object drop

Indicates if you want ZDT/Db2 to display a confirmation panel each time you enter a DR(op) command against an object

Parent panels

- Utility Functions panel on page 915

Child panels

To display this panel...	Use/do this
Create Database panel on page 529	Select object type **Database** and enter `C`
Create Table Space panel on page 564	Select object type **Table space** and enter `C`
Create Table panel on page 561	Select object type **Table** and enter `C`
Create View panel on page 601	Select object type **View** and enter `C`
Create Alias panel on page 527	Select object type **Alias** and enter `C`
Create Index panel on page 542	Select object type **Index** and enter `C`
Create Synonym panel on page 560	Select object type **Syntax** and enter `C`
Create Distinct Type panel on page 531	Select object type **Distinct type** and enter `C`
Create Function panel on page 536	Select object type **Function** and enter `C`
Create Procedure panel on page 553	Select object type **Stored procedure** and enter `C`
Create Trigger panel on page 596	Select object type **Trigger** and enter `C`
Create Auxiliary Table panel on page 528	Select object type **Auxiliary table** and enter `C`
Select object type **Database** and enter `D`	
Select object type **Table space** and enter `D`	
Select object type **Table** and enter `D`	
Select object type **View** and enter `D`	
Select object type **Alias** and enter `D`	
Select object type **Index** and enter `D`	
Select object type **Syntax** and enter `D`	

To display this panel...	Use/do this
Select object type **Distinct type** and enter `D`	
Select object type **Function** and enter `D`	
Select object type **Stored procedure** and enter `D`	
Select object type **Trigger** and enter `D`	

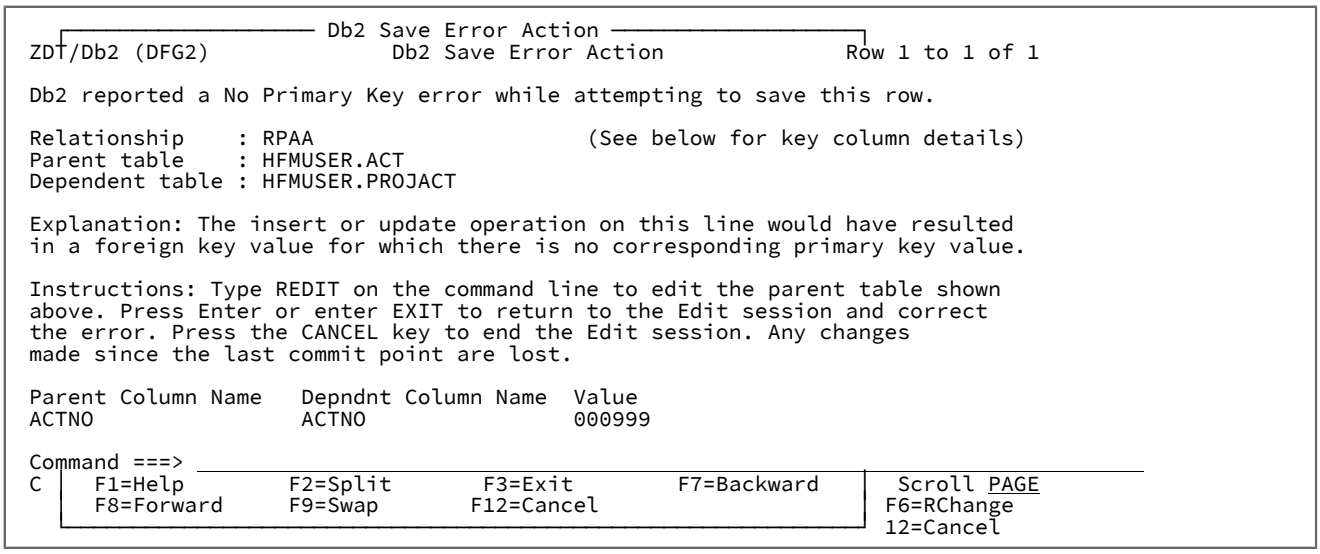
Related tasks

- Creating and dropping Db2 objects on page 208

Db2® Save Error Action panel

You use the Db2® Save Error Action panel to determine the type of error, and what action you can take, when an error occurs during a Z Data Tools/Db2 Edit session.

Panel and field definitions



Related tasks

- Example of correcting an error during an Edit session on page 205

Db2® Subsystem Selection panel

You use the Db2® Subsystem Selection menu to select the Db2® subsystem to which you want ZDT/Db2 to connect.

Panel and field definitions

```
 Process   Help
_____________________________________________________________________________
ZDT/Db2 (DFG1)             Db2 Subsystem Selection                Row 1 of 38
Command ===>                                                      Scroll CSR

Sel SSID Status    Description                                    Prefix
-------------------------------------------------------------------------
 _  DFG1 ACTIVE    Db2 Version 12 group with DFG1                 -DFG1
 _  DFH1 ACTIVE    Db2 Version 12 group with DFH1                 -DFH1
 _  DF31 ACTIVE    Db2 Version 13.1 - FMD1 DF31                   -DF31
    DF11 INACTIVE  Db2 Version 12 - FMD1 DF11                     -DF11

 F1=Help      F2=Split     F3=Exit      F5=Refresh   F6=Show      F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Note: The Db2® Subsystem Selection panel shown here shows an inactive Db2® subsystem. Your systems administrator might have customized the list so that, for example, only active Db2® subsystems appear. For more information, see your systems administrator or see the *Z Data Tools Customization Guide*.

The fields on the Db2® Subsystem Selection menu are:

Sel

Selection field. Enter `s` or `/` to select the Db2® subsystem shown on that line.

SSID

The subsystem ID. This ID is displayed in the **Db2® SSID** field on the Primary Option Menu and shows to which Db2® subsystem ZDT/Db2 is currently connected.

Status

Shows status of the Db2® subsystem:

ACTIVE

Db2® system is active and is available for connection.

GROUP

Db2® system is in a Db2® data sharing environment and is available for connection.

INACTIVE

Db2® system is not active and is not available for connection.

UNAVAIL.

ZDT/Db2 has not been installed on a subsystem or Db2® data sharing group, or ZDT/Db2 is not available for connection.

Description

The descriptive name of the Db2® subsystem.

Prefix

Command prefix for the Db2® subsystem.

Parent panels

- Primary Option Menu panel on page 760

Child panels

None.

Related tasks

- Selecting the Db2 subsystem when more than one is available on page 30

Db2® Template Description panel

You use the Db2® Template Description panel to add a description to the template you are creating or modifying.

Panel and field definitions

```
                        Db2 Template Description
s                                                                          s
F
C   Template
T      Data set . : SHRIKE2.HFM.TEMPLATE(EMP)
-
1   Source
       Db2 Object : EMP                                                    D
C      Subsystem  : DFG2    Release : 1015

    Created . . . : 2013/08/09 12:40:22

    Modified  . . : 2013/09/12 11:59:45

    Description . . _____________________________________________________
                  . _____________________________________________________
                  . _____________________________________________________
                  . _____________________________________________________
                  . _____________________________________________________

    Command ===> _____________________________________________________
C    F1=Help       F2=Split      F3=Exit       F7=Backward   F8=Forward
     F9=Swap      F12=Cancel
```

Template Data set

The name of the partitioned data set where the template is stored.

Db2® Object

The name of the Db2® object from which the template was built.

Subsystem

The subsystem ID of the Db2® subsystem where the Db2® object resides.

Release

The Db2® version, release, and modification level of the Db2® subsystem on which the object resides.

Created

The date that the template was created from the copybook.

Modified

The date that the template was last modified.

Description

The description of the template. Use this field to enter or modify the description of the template.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

Related tasks

- Editing a template on page 84

Db2® Utilities panel

You use the Db2® Utilities panel to generate Db2® utility jobs to copy, load, rebuild, reorganize and collect statistics on Db2® objects.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                  Db2 Utilities

Processing Options:
   Db2 Utility:   (valid obj)   Object Type:          Specification:
   1  1. COPY        (2,3)      2  1. Tables         1  1. Object name
      2. LOAD        (1)           2. Table spaces      2. LISTDEF
      3. REBUILD     (4,5)         3. Index spaces
      4. RECOVER     (2,3)         4. Indexes
      5. REORG       (2,4)         5. Indexes for table spaces
      6. RUNSTATS    (2,4)
      7. UNLOAD      (1,2)

Db2 Object Details:             (* for list in any field)
   Database  . . . . . ________
   Table/Index space . ________

   Table or Index:
   Owner . . . . . . . _______________  +
   Name  . . . . . . . ____________________________________________  +

LISTDEF Details:                (* unavailable)
   LISTDEF name  . . .  ________________
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

The Db2® Utilities panel shows a list of the available utilities on the left side of the *"Processing Options"* section, and a list of Db2® object types on the right. Each Db2® utility applies to only certain object types. For example, the LOAD utility only applies to Db2® tables. The valid object types for each utility are shown under the heading *"Valid Db2® Obj"*.

The panel shows a **Specification** section and a **LISTDEF name** field. A selection of LISTDEF means that a LISTDEF name is to be specified as the object of the utility; otherwise, the object name of the type specified in object type is to be specified.

In the **Specification** section, you can select:

1. Object name

The object specified in **Db2® Object Details** is used as the object of the utility. The default is 2 (**Table spaces**).

2. LISTDEF

The name specified in **LISTDEF name** is used as the object of the utility. The default is 1 (**Object name**).

Db2® Utility

Specifies the Db2® utility to be executed. The default is 1 (**Copy**).

Note: The LOAD syntax does not have a LISTDEF clause. If you select LOAD in the **Db2® Utility** section and LISTDEF in the **Specification** section, ZDT/Db2 issues a warning message.

In the **LISTDEF name** section, if you have specified `2` (LISTDEF) in the **Specification** section, specify the name of the LISTDEF you want to use.

You specify the objects to be processed by the Db2® utility at the bottom of the panel.

Database

The name of the database for the object.

Table/Index space

The name of the table space or index space for the object.

Owner

The name of the owner of the table or index for the object.

Name

The name of the table or index for the object.

Parent panels

- Primary Option Menu panel on page 760
- Table Spaces panel on page 879
- Tables, Views and Aliases panel on page 891
- Indexes panel on page 723

Child panels

To display this panel...	Use/do this
COPY Utility (Table Spaces) panel on page 524	Db2® utility: COPY; object type: Table space; specification: Object name
COPY Utility (Table Spaces) with LISTDEF panel on page 526	Db2® utility: COPY; object type: Table space; specification: LISTDEF
COPY Utility (Index Spaces) panel on page 517	Db2® utility: COPY; object type: Index space; specification: Object name
COPY Utility (Index Spaces) with LISTDEF panel on page 519	Db2® utility: COPY; object type: Index space; specification: LISTDEF
LOAD Utility panel on page 733	Db2® utility: LOAD; object type: Tables; specification: Object name
LOAD Utility panel using Db2 templates on page 734	Db2® utility: LOAD; object type: Tables; specification: LISTDEF
REBUILD Utility (Indexes) panel on page 777	Db2® utility: REBUILD; object type: Indexes; specification: Object name

To display this panel…	Use/do this
REBUILD (Indexes) with LISTDEF panel on page 779	Db2® utility: REBUILD; object type: Indexes; specification: LISTDEF
REBUILD (Indexes for Table Spaces) panel on page 773	Db2® utility: REBUILD; object type: Indexes for Table space; specification: Object name
RECOVER Utility (Table Spaces) panel on page 783	Db2® utility: RECOVER; object type: Table space; specification: Object name
RECOVER Utility (Table Spaces) with LISTDEF panel on page 785	Db2® utility: RECOVER; object type: Table space; specification: LISTDEF
RECOVER Utility (Index Spaces) panel on page 780	Db2® utility: RECOVER: object type: Index spaces; specification: Object name
REORG Utility (Table Spaces) panel on page 795	Db2® utility: REORG; object type: Table space; specification: Object name
REORG Utility (Indexes) panel on page 792	Db2® utility: REORG; object type: Indexes; specification: Object name
REORG Utility (Indexes) with LISTDEF panel on page 793	Db2® utility: REORG; object type: Indexes; specification: LISTDEF
RUNSTATS Utility (Table Spaces) panel on page 808	Db2® utility: RUNSTATS; object type: Table space; specification: Object name
RUNSTATS Utility (Table Spaces) with LISTDEF panel on page 810	Db2® utility: RUNSTATS; object type: Table space; specification: LISTDEF
RUNSTATS Utility (Indexes) panel on page 805	Db2® utility: RUNSTATS; object type: Indexes; specification: Object name
RUNSTATS Utility (Indexes) with LISTDEF panel on page 807	Db2® utility: RUNSTATS; object type: Indexes; specification: LISTDEF

To display this panel…	Use/do this
UNLOAD Utility (Tables) "From" panel on page 910	Db2® utility: UNLOAD; object type: Tables; specification: Object name
UNLOAD Utility (Table Spaces) panel on page 908	Db2® utility: UNLOAD; object type: Table spaces; specification: Object name
UNLOAD Utility (Table Spaces) with LISTDEF panel on page 909	Db2® utility: UNLOAD; object type: Table spaces; specification: LISTDEF

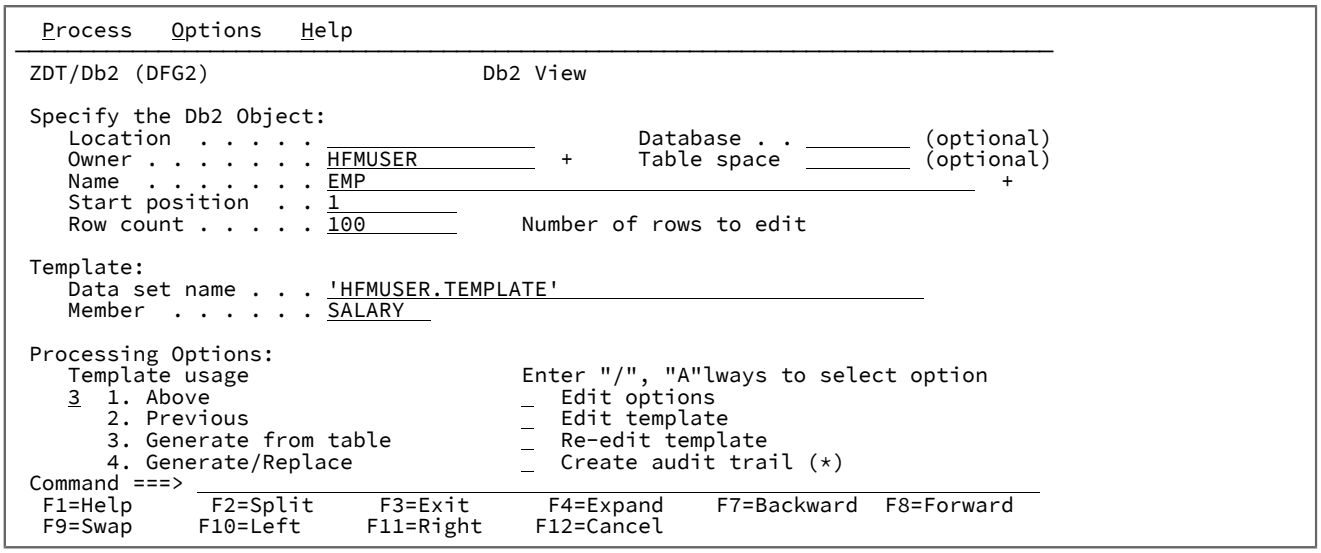
Related tasks

- Generating batch JCL for Db2 utility jobs on page 346

Db2® View panel

You use the Db2® View panel to enter the name of an object to be edited (but without the ability to save any changes you make). You also use this panel to specify the name of a template that describes a logical view of the object.

Panel and field definitions



Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Start position

Indicates where ZDT/Db2 is to begin loading rows from the result table into the ZDT/Db2 editor.

If you specify a value greater than 1 (the default), ZDT/Db2 skips n-1 rows of the result table before loading rows into the ZDT/Db2 editor. If the **Row count** value is `ALL` (or `0` or `*`), when the data is displayed you can scroll up to see the rows that were skipped. For a fixed **Row count** value, you cannot scroll backwards to see any skipped rows.

Row count

Use this option to specify the maximum number of rows to be retrieved from Db2® and loaded into memory for the editor session. You can specify a value in the range 1–999999999 or, to indicate that all rows are to be retrieved, specify `ALL`, `0`, or `*`. The default value is 100.

A Z Data Tools/Db2 editor session only applies to those rows fetched from Db2® and loaded into the editor. It does not apply to any unfetched rows. This means that the FIND and CHANGE primary commands only apply to those rows fetched from Db2®.

Note: When you are editing large tables, use the value of 0 with caution as it is possible to encounter storage problems. The amount of storage required by the editor depends on both the number of rows retrieved, and the row length. Use the row selection criteria section of the template to limit the number of rows retrieved.

Data set name

The name of a data set that contains, or will contain, the template. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The Db2® View panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. Generate from table

Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select this option to change the options for the current editor session:

Edit options

Enter one of these:

A / or an "A"

Displays the first ZDT/Db2 editor options panel, prior to displaying the data for the Db2® object being processed.

A number in the range 1–8

Displays the *n*th ZDT/Db2 editor options panel directly where *n* is the value entered.

You can select the following options by entering a / or an "A":

Edit template

Select this option to edit the template before editing the data.

Re-edit template

Select this option to control the navigation between panels within the Edit dialog. This option controls the behavior of the F3 and F12 keys when pressed from within an Edit session.

Create audit trail



Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be

changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Parent panels

- Primary Option Menu panel on page 760
- Tables, Views and Aliases panel on page 891

Child panels

To display this panel…	Use/do this
Table View panel on page 882	No options selected and press Enter
Editor Options (1 of 8) panel on page 644	Select **Edit options**
Column Selection/Edit panel on page 497	Select **Edit template** or **Re-edit template**

Related tasks

- Viewing and changing Db2 data on page 122
- Selecting options on ZDT/Db2 panels on page 45
- Specifying a data set and a member name on page 52
- Where you can use templates on page 22
- Working with templates on page 74
- View and Edit options (options 1 and 2) on page 67

Db2I Primary Option Menu panel

You use the Db2I Primary Option Menu panel to interface to Db2® Interactive.

Panel and field definitions

```
                          Db2I PRIMARY OPTION MENU          SSID: DFA2
COMMAND ===>

Select one of the following Db2 functions and press ENTER.

 1  SPUFI                  (Process SQL statements)
 2  DCLGEN                 (Generate SQL and source language declarations)
 3  PROGRAM PREPARATION    (Prepare a Db2 application program to run)
 4  PRECOMPILE             (Invoke Db2 precompiler)
 5  BIND/REBIND/FREE       (BIND, REBIND, or FREE plans or packages)
 6  RUN                    (RUN an SQL program)
 7  Db2 COMMANDS           (Issue Db2 commands)
 8  UTILITIES              (Invoke Db2 utilities)
 D  Db2I DEFAULTS          (Set global parameters)
 X  EXIT                   (Leave Db2I)




 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Parent panels

- Primary Option Menu panel on page 760

Child panels

None.

Related tasks

- Interfacing with Db2 Interactive on page 438

Details panels

These panels are:

- Database Details panel
- Table Space Details panel
- Table Details panel
- View Details panel
- Materialized Query Table Details panel [26]
- Index Details panel
- Column Details panel
- Synonym Details panel
- Application Plan Details panel

26. The View Details panel contains some specific details for Materialized Query objects.

- Package Details panel
- DBRM Details panel
- Distinct Type Details panel
- Function Details panel
- Procedure Details panel
- Trigger Details panel

ZDT/Db2 displays one of these panels when you specify the I line command against an object displayed in a list of Db2® objects.

Child panels

None.

Display Row panel

ZDT/Db2 displays the Display Row panel if you use the ROW line command on any panel where you can use the ROW line command.

Related tasks

Distinct Types panel

You use the Distinct Types panel to list the distinct types in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSDATATYPES.

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)                  Distinct Types
Location:                                                    Format TABL

                                                                        +
SEL    SCHEMA  NAME     OWNER    CREATEDBY SOURCESCHEMA SOURCETYPE METATYEX
       *       *        *        *         *            *          *
----   #1--+-- #3--+--- #2--+--- #4--+---- #5--+----1-- #6--+----- #18-+----1-
****  Top of data  ****
_____  SHRIKE2 DT1      SHRIKE2  SHRIKE2   SYSIBM       CHAR       Distinct Ty
_____  SHRIKES CHAR80   SHRIKE2  SHRIKE2   SYSIBM       CHAR       Distinct Ty
****  End of data  ****









Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Schemas panel on page 812
- Functions panel on page 709

Child panels

To display this panel...	Use/do this
See Comment panels on page 506	Line command COM
Create Distinct Type panel on page 531	Line command CR
See Drop panels on page 640	Line command DR
Grant privileges panels on page 715	Line command G
	Line command GEN

To display this panel…	Use/do this
Generate SQL From Db2 Catalog panel on page 712	
	Line command I
Details panels on page 636	
	Line command P
Privileges panels on page 768	
Line command PS	
	Line command R
Revoke privileges panels on page 798	
	Line command ROW
Display Row panel on page 637	
	Line command SCH
Schemas panel on page 812	
	Line command T
Tables, Views and Aliases panel on page 891	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Drop panels

These panels are:

- Drop Alias panel
- Drop Database panel
- Drop Distinct Type panel
- Drop Function panel
- Drop Index panel
- Drop Stored Procedure panel
- Drop Synonym Procedure panel
- Drop Table panel
- Drop Table Space panel
- Drop Trigger panel
- Drop View panel

ZDT/Db2 displays one of these panels when you specify the DR line command against an object displayed in a list of Db2® objects.

You use the appropriate Drop panel when you want to drop a Db2® object.

If you have checked the **Confirm object drop** option on the Db2® Object Functions panel, the Object List Utility panel, or the Explain Utilities panel, before dropping the Db2® object ZDT/Db2 displays the Drop Confirmation panel.

Child panels

- Drop Confirmation panel on page 640

Related tasks

- Dropping Db2 objects on page 243

Drop Confirmation panel

ZDT/Db2 displays the Drop Confirmation panel when you have specified that you want to drop a Db2® object *and* you have checked the **Confirm object drop** option on the Db2® Object Functions panel or the Explain Utilities panel.

To proceed with the drop, press Enter; to cancel the drop, press the Cancel function key (F12).

Panel and field definitions

```
  Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                Drop Confirmation

Table  : "ADDRESS BOOK"

Instruction:
  Press Enter to confirm drop.
  (The table will be dropped)
  (Warning: All dependent objects will also be dropped.)

   Press CANCEL or EXIT to cancel drop.

   Type SQL to generate the SQL for the drop statement only.




Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Parent panels

- Db2 Object Functions panel on page 622
- Db2 Object Functions panel on page 622

Child panels

None.

Related tasks

- Dropping Db2 objects on page 243

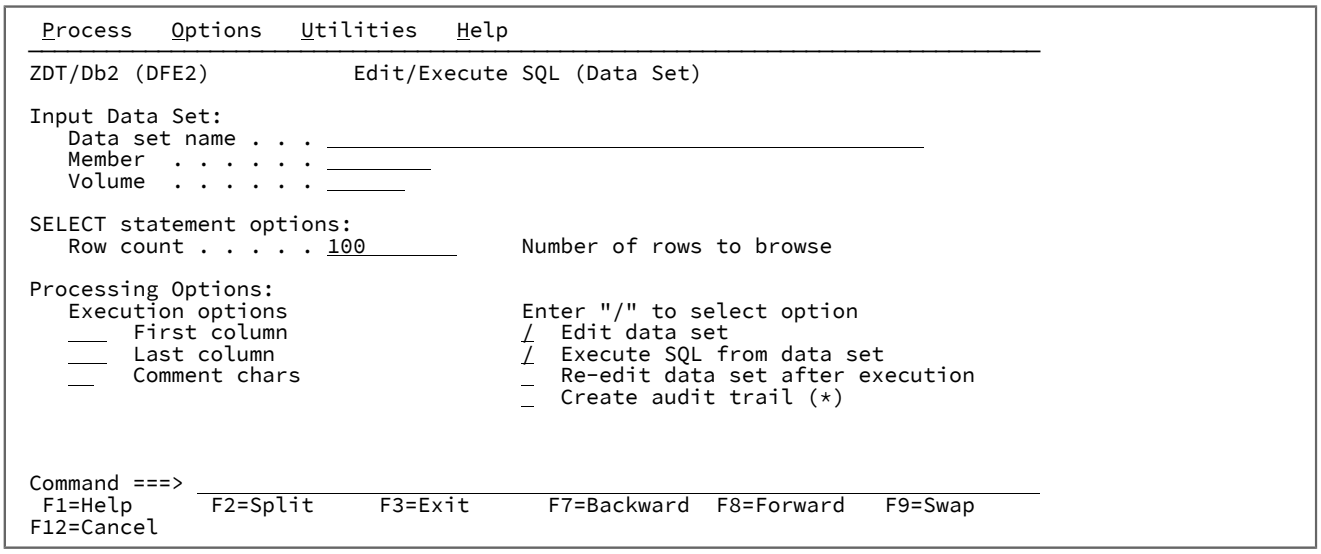
Related references

- Db2 Object Functions panel on page 622

Edit/Execute SQL (Data Set) panel

You use the Edit/Execute SQL (Data Set) panel to edit and execute SQL statements contained in a data set.

Panel and field definitions



Data set name

The name of the data set or PDS member containing the SQL statements you want to edit. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial

The serial number of the volume that contains the data set (if applicable).

Row count

Limits the rows loaded into the ZDT/Db2 editor when the SQL statement being processed is a SELECT statement. To load all rows for the SELECT statement, specify "*" or ALL in this field.

First column

First data column. When generating SQL statements, ZDT/Db2 ignores any data to the left of this column number. If you do not specify a value, the statements are assumed to start in column 1.

Last column

Last data column. When generating SQL statements, ZDT/Db2 ignores any data to the right of this column number. If you do not specify a value, the statements are assumed to continue to the end of the data set record.

Comment chars

Use this option to specify one or two characters (for example "--") that denote records in the data set that are comment lines. If the specified characters appear as the first nonblank characters in a data set record (and that part of the statement is not in a quoted string), ZDT/Db2 ignores the record.

Edit data set

ZDT/Db2 starts an ISPF Edit session for the specified member or data set.

Use this option to use a library of SQL statements that you can maintain ready for execution.

To leave the Edit session, use the Exit function key (F3). Any changes you have made are saved.

Execute SQL from data set

ZDT/Db2 reads the data set or member, executes the SQL statements that it can extract, returning the results if applicable.

To execute the next SQL statement in the data set or member after ZDT/Db2 displays a result table or error panel, use the Exit function key (F3).

To return to the Edit and Execute SQL Statements from a Data Set panel without executing the statements, or at any time when an error or result panel has been displayed, use the Cancel function key (F12).

Re-edit data set after execution

After ZDT/Db2 has completed executing the statements in the data set, it restarts the Edit session (instead of returning to the Edit and Execute SQL Statements from a Data Set panel).

Create audit trail

Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Parent panels

- SQL Prototyping, Execution and Analysis panel on page 843

Child panels

None.

Related tasks

- Editing and executing SQL statements from a data set on page 429
- Specifying a data set and a member name on page 52
- SAF-rule controlled auditing on page 71

Editor Options (1 of 8) panel

Use the eight Editor Options panels (1 to 8) to set your default editor options.

If you get to this panel from the Set Processing Options panel or from the action bar, these options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Otherwise (if you get to this panel by selecting **Edit options** on the Db2® View panel or the Db2® Edit panel), then any changes you make to these options last only for the life of the current ZDT/Db2 editor session.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)             Editor Options (1 of 8)             Global Settings

Display Format:                        Editor CAPS Setting:
   Initial display format                 Enter "/" to select option
   1  1. Previous                         /  Initially ON
      2. Table
      3. Single                        Audit Option:
                                          Enter "/" to select option
                                          /  Create audit trail
More Options:
   Enter "/" to select option
   /  Optimization, large table and data sampling options
   /  Table/Single format, prefix area and shadow options
   /  Key, index and export options
   /  Varying length and nullable column options
   /  Enter key and commit processing options
   /  Concurrency and locking options
   /  XML and LOB column processing options

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F11=NxtPage  F12=Cancel
```

Initial display format

Use this section of the Editor Options panel to specify how editor panels appear at the start of a Z Data Tools/Db2 editor session. Set this option to one of the following:

1. **Previous**

   The display format field (**Format**) on the View or Edit panel is prefilled with the value specified in the previous ZDT/Db2 editor session. This is the default setting.

2. **Table**

   The View or Edit panel is displayed in TABLE display format.

3. **Single**

   The View or Edit panel is displayed in SNGL display format.

Editor CAPS Setting

Indicates the initial setting of CAPS in a Z Data Tools/Db2 editor session and whether you can change the setting. When this option is available for update, the default is not selected.

Initially ON (fixed at installation)

CAPS set to ON. You cannot change the setting by means of the edit options.

Initially OFF (fixed at installation)

CAPS set to OFF. You cannot change the setting by means of the edit options.

Enter "/" to select option … Initially ON

You can change the CAPS setting:

`/`

Set CAPS to ON

(blank)

Set CAPS to OFF

Create audit trail

Determines if ZDT/Db2 generates an audit trail report of all successful modifications to Db2® data made during an Edit session:

/

Audit trail report generated.

(blank)

Audit trail report not generated.

Note:

1. The display of this option depends on whether SAF-rule controlled auditing is in effect and if auditing is to occur for the Db2® system. See SAF-rule controlled auditing on page 71.

2. The ability to change this option depends on installation options within the Db2® subsystem to which ZDT/Db2 is connected. The **Create an audit trail** option can be:
   - Permanently selected ("/" shown in the option). Audit trail reporting occurs for Edit processing. You cannot change the setting of the option.
   - Permanently deselected (blank shown in the option). Audit trail reporting does not occur for Edit processing. You cannot change the setting of the option.
   - Optional. Type a "/" in the option to generate audit reporting for all Edit processing; otherwise, type a space in the option to suppress audit trail reporting of any Edit processing.

   At the end of your ZDT/Db2 session, the **Create an audit trail** option that you specify can be saved to your ISPF profile, or it can be set to blank (audit trail report not selected), depending on the installation options within the Db2® subsystem to which ZDT/Db2 is connected. The audit installation options for the currently connected Db2® system can be determined by selecting option 7 from the HELP pull down menu. When the auditing option is *"OPTIONAL,OFF"*, the Create an audit trail option is always reset to blank (not selected) at the start of a Z Data Tools/Db2 session. This means that, for an audit log to be written, auditing must be explicitly selected each time a new ZDT/Db2 session is started. Further, when this option is selected, there is no distinction between the global setting for the audit option, and any value entered using 'edit options' on the Db2® edit panel. For more information, see the *Z Data Tools Customization Guide* or see your systems administrator.

More options

These options control whether ZDT/Db2 displays the further editor options panels. The default for each option is selected.

The ZDT/Db2 editor options are too numerous to fit on a single panel. To access additional options, select the fields in this section of the panel.

To navigate to the other options panels, press the PF key assigned to NxtPage (PF11 by default). To return to the previous panel, press the PF key assigned to PrvPage (PF10 by default).

You can omit the display of one or more of the additional options panels by de-selecting the appropriate field in the "More options" section of the panel.

Optimization, large table and data sampling options

Indicates whether you want the second Editor Options panel displayed when you navigate between Editor Options panels using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the second Editor Options panel

(blank)

Do not display the second Editor Options panel

Table/Single format, prefix area and shadow options

Indicates whether you want the third Editor Options panel displayed when you navigate between Editor Options panels using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the third Editor Options panel

(blank)

Do not display the third Editor Options panel

Key, index and export options

Indicates whether you want the fourth Editor Options panel displayed when you navigate between Editor Options panels using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the fourth Editor Options panel

(blank)

Do not display the fourth Editor Options panel

Varying length and nullable column options

Indicates whether you want the fifth Editor Options panel displayed when you navigate between Editor Options panels using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the fifth Editor Options panel

(blank)

Do not display the fifth Editor Options panel

Enter key and commit processing options

Indicates whether you want the sixth Editor Options panel displayed when you navigate between Editor Options panels using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the sixth Editor Options panel

(blank)

Do not display the sixth Editor Options panel

Concurrency and locking options

Indicates whether you want the seventh Editor Options panel displayed when you navigate between Editor Options panels using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the seventh Editor Options panel

(blank)

Do not display the seventh Editor Options panel

XML and LOB column processing options

Indicates whether you want to display the eighth Editor Options panel when you navigate between Editor Options panels by using the NxtPage function key (F11) or the PrvPage function key (F10).

/

Display the eighth Editor Options panel

(blank)

Do not display the eighth Editor Options panel

Parent panels

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Edit panel on page 616
- Db2 Browse panel on page 612
- Action bar pull-down menu on page 445

Child panels

To display this panel…	Use/do this
Editor Options (2 of 8) panel on page 649	Press the NxtPage function key (F11)

Related tasks

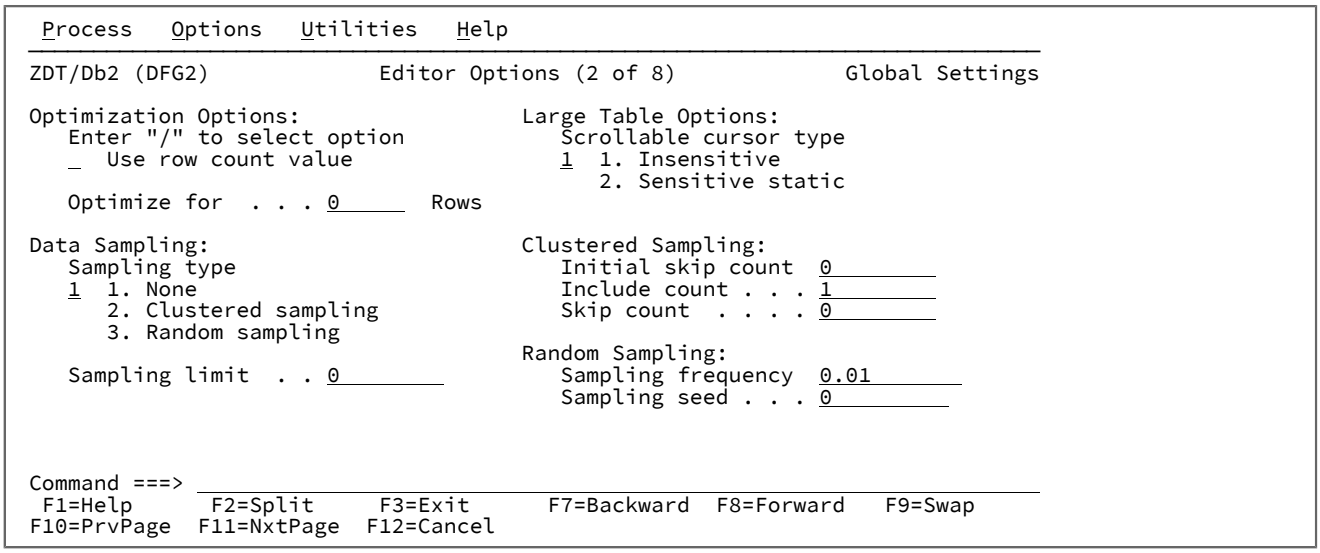
- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126
- Selecting a display format on page 153
- Installation settings that affect CAPS on page 180

Editor Options (2 of 8) panel

You use the Editor Options panel (2 of 7) to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions



Use row count value

This option influences Db2® optimization.

When ZDT/Db2 prepares an SQL statement to access Db2® data, the number of rows in the result table may affect the method used by Db2® to produce the result table. In some cases, Db2® uses a different, more efficient, access method if it is known that only a few rows of an otherwise very large result table are required. This option enables an additional clause to be added to the SQL statement used by ZDT/Db2 when accessing Db2®. The OPTIMIZE FOR *nnn* ROWS clause is documented in the *SQL Reference* manual for the appropriate version of Db2®. The default is selected.

Enter a number in the range 0-999999 inclusive. This option and the **Use row count value** option combine in these ways:

Use row count value option selected

The value specified for **Row count** on the editor function entry panel is used, unless ALL (also 0, *) is specified. In the latter case, the **Optimize for** value is considered. A zero value results in no a OPTIMIZE FOR clause. Any other value results in an OPTIMIZE FOR *nnn* ROWS clause being added. *nnn* is the value entered in the **Optimize for** option. The default for the **Optimize for** option is 0.

Use row count value option not selected

The **Optimize for** value is considered. A zero value results in a no OPTIMIZE FOR clause. Any other value results in an OPTIMIZE FOR *nnn* ROWS clause being added. *nnn* is the value entered in the **Optimize for** option

Scrollable cursor type

Controls the access method used by ZDT/Db2 when processing large Db2® objects.

The ZDT/Db2 editor can operate in two modes. When there is sufficient memory available, ZDT/Db2 loads all of the rows for the result table into the editor. For Db2® objects with large result tables, ZDT/Db2 loads only a small number of rows initially and keeps only a small number of rows in memory during the editor session. This is achieved using Db2® scrollable cursors. The type of cursor specified can influence the behavior of the editor, particularly in edit mode. The default is 1 (**Insensitive**).

1. **Insensitive**

   An insensitive cursor operates like a point-in-time snapshot of the result table. The number of rows in the result table is fixed, and is not changed by insert, update, or delete operations made to the table by other processes. ZDT/Db2 always uses an insensitive cursor in browse or view modes.

2. **Sensitive static**

   A sensitive static cursor also has a fixed number of rows in the result table, however delete and update operations made using the cursor are reflected the next time the row is processed by the cursor. In addition, update and delete operations made to the underlying table may cause the row to be no longer visible when processed by the cursor.

   A sensitive static cursor is only used in edit mode.

Sampling type

Controls whether data sampling is used.

Data sampling enables a subset of the rows in a Db2® object to be loaded into the editor. The default is 1 (**None**). The possible values for this option are:

1. **None**

   All rows of the object are eligible for inclusion, this is the default.

2. **Clustered sampling**

   A specified number of rows are loaded into the editor.

3. **Random sampling**

   Each row of the object is considered for inclusion in the editor. A random number generator and a frequency are used to determine whether the row is actually loaded.

When data sampling is selected, you also need to consider these options:

Sampling limit

For both clustered and random sampling.

Initial skip count

For clustered sampling.

Include count

For clustered sampling.

Skip count

For clustered sampling.

Sampling frequency

For random sampling.

Sampling seed

For random sampling.

Clustered Sampling

Clustered data sampling divides the data into consecutive groups of rows. The number of rows in each group is the sum of these values:

- Initial skip count
- Include count
- (Final) skip count

Within each group, the number of rows specified in **Initial skip count** are ignored. The number of rows in **Include count** are sampled (added to the editor session). Finally, the number of rows specified in the **Skip count** are ignored. Sampling then recommences with the next group of rows.

These options apply only to clustered data sampling. Any values specified are ignored when clustered data sampling is not in use.

Initial skip count

Controls how many rows are initially skipped in each clustered sample group. It can be zero, or a positive integer. The default is 0.

Include count

Controls how many rows are sampled in each clustered sample group. It must be a positive integer. The default is 1.

Skip count

Controls how many rows are skipped at the end of each clustered sample group. It can be zero, or a positive integer. The default is 0.

Random Sampling

Random data sampling uses a random number generator, together with the sampling frequency to determine whether a row is included in the sample. These options apply only to random data sampling. Any values specified are ignored when random data sampling is not in use.

Sampling limit

A value that sets a maximum for the number of rows in the data sample. A zero value indicates that there is no limit on the number of rows in the data sample. The default is 0.

Sampling frequency

Controls the sampling frequency for random data sampling. The default is 0.01.

Enter a value in the range 0.000000001 to 0.999999999 inclusive. To sample approximately 10% of rows enter a frequency of 0.1. To sample approximately 1% of rows enter a frequency of 0.01.

Sampling seed

A value used to initiate the random number generator used for random data sampling. The default is 0.

Enter a integer in the range 0 to 2147483647 inclusive. A zero value indicates that ZDT/Db2 should use the system clock to initiate the random number generator. Specify a zero value, when repeatability of data samples is not required. Any non-zero value used to initiate the random number generator ensures that data samples are repeatable, given the same table and other input conditions.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (2 of 8) panel, you then press the NxtPage function key (F11).

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (2 of 8) panel if the **Optimization, large table and data sampling options** option on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel...	Use/do this
Editor Options (1 of 8) panel on page 644	Press the PrvPage function key (F10)
Editor Options (3 of 8) panel on page 655	Press the NxtPage function key (F11)

Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126
- Data sampling on page 170
- Large table options on page 170

Editor Options (3 of 8) panel

You use the Editor Options (3 of 8) panel to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)             Editor Options (3 of 8)             Global Settings

Table Display Format:                 Prefix Area:
   Enter "/" to select option            Prefix length . . . A  (A,6,7,8,9)
   /  Show data type
   /  Show scale line                    Enter "/" to select option
   /  Show column number                 /  Display prefix area
                                         _  Show prefix area on the right
Single Display Format:                   _  Show SQLCODE
   Enter "/" to select option
   _  Show data type                  Shadow Lines:
   /  Show keys, indexes                 Enter "/" to select option
   _  Show column number                 /  Show excluded shadow lines
   _  Left justify numerics
   _  Show char position range



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F11=NxtPage  F12=Cancel
```

Table Display Format Options

Use this section of the Editor Options panel to specify the information you want displayed in table display format (TABL):

Show data type

Indicates if you want ZDT/Db2 to display data type information. The default is selected.

/

Show the data type for each column displayed in table display format.

ZDT/Db2 displays an additional line (immediately below the line containing the column reference numbers, if displayed) showing the data type for each column.

(blank)

Do not show data type information.

Show scale line

Indicates if you want ZDT/Db2 to display scale line information. The default is selected.

/

Show a scale line for each column displayed in table display format.

The scale shows the left and right boundary of the data for the column and, for non-numeric data types, positioning information.

ZDT/Db2 displays the scale line immediately above the first line of data. The scale line also displays information about primary and foreign keys, and unique indexes.

(blank)

Do not show scale line information.

Show column number

Indicates if you want ZDT/Db2 to display column number information. The default is selected.

/

Show the column number (as defined in the Db2® catalog) for each column displayed in table display format.

ZDT/Db2 displays the column numbers immediately below the line containing the column names.

(blank)

Do not show column number information.

Prefix Area

These options controls the prefix area in a Z Data Tools/Db2 editor session.

Prefix length

Controls the width of the prefix area in a Z Data Tools/Db2 editor session.

The ZDT/Db2 editor can display a prefix area on either the left, or the right of the screen. The width of the prefix area can be 6, 7, 8 or 9 characters. The default is 6, specified by typing A in the field. You can type a different value to set the width of the prefix area to your preference. Note that the prefix area does not appear in a Z Data Tools/Db2 browse session, and only appears in a Z Data Tools/Db2 view or edit session when the **Display prefix area** option is selected. You can alter the prefix length from within an editor session using the PREFIX editor primary command.

Display prefix area

Controls whether the prefix area is displayed in a Z Data Tools/Db2 edit or view session. The default is selected.

The ZDT/Db2 editor can optionally display a prefix area, where prefix commands may be entered.

/

Display the prefix area.

(blank)

Do not display the prefix area.

You can turn the prefix area on or off from within a Z Data Tools/Db2 editor session (edit and view only) by issuing the PREFIX editor primary command.

Show prefix area on the right

Controls the location of the ZDT/Db2 editor's prefix area. The default is not selected.

The ZDT/Db2 editor can display the prefix area on either the left, or the right of the screen.

/

Display the prefix area on the right of the screen.

(blank)

Display the prefix area on the left of the screen.



Note:

1. The prefix area does not appear in a Z Data Tools/Db2 browse session.
2. This option has no effect unless the **Display prefix area** option is selected.
3. You can move the prefix area, from within the ZDT/Db2 editor, by issuing the PREFIX LEFT or PREFIX RIGHT editor primary commands.

Show SQLCODE

Controls whether the ZDT/Db2 editor shows the SQLCODE in the prefix area, when an insert, update, or delete operation for a row fails. The default is not selected.

ZDT/Db2 processes changes to data in the edit session one row at a time. If Db2® returns an error when the change is presented to Db2®, ZDT/Db2 shows either `=ERR` or an SQLCODE in the prefix area for the row.

/

ZDT/Db2 editor shows the SQLCODE in the prefix area.

(blank)

ZDT/Db2 editor shows the `=ERR` in the prefix area.

Single display Format Options

Use this section of the Editor Options panel to specify the information you want displayed in single display format (SNGL).

Show data type

Indicates if you want ZDT/Db2 to display data type information. The default is not selected.

`/`

Show the data type for each column displayed in single display format.

ZDT/Db2 displays an additional column (immediately to the right of the column containing the name of the column) showing the data type for each column.



Note: When you are displaying data in single display mode, you can toggle the display of data type information by using the TYPE primary command.

(blank)

Do not show data type information.

Show keys, indexes

Indicates if you want ZDT/Db2 to display information showing if a column is a part of a key or index. The default is selected.

`/`

Show an additional column, 4 characters wide, indicating if the column is part of a key or index.

Note: When you are displaying data in single display mode, you can toggle the display of index information by using the INDEXINF primary command.

(blank)

Do not show key or index information.

Show column number

Indicates if you want ZDT/Db2 to display column number information. The default is not selected.

/

Show the column number (as defined in the Db2® catalog) for each column displayed in single display format.

ZDT/Db2 displays an additional column (the leftmost on the panel) containing the column numbers.

Note:

1. When you are displaying data in single display mode, you can toggle the display of column numbers by using the REFS primary command.
2. In single display mode, when ZDT/Db2 displays the column number, it also displays the column heading *"Rdf"* (as the second column heading from the left). The *"Rdf"* column is unused by ZDT/Db2 and can be ignored.

(blank)

Do not show column number information.

Left justify numerics

Indicates if you want ZDT/Db2 to display numeric data left-justified, in single display format. The default is not selected.

/

Data for numeric columns is displayed left-justified. That is, the first characters of any numbers are shown at the extreme left of the column.

(blank)

Data for numeric columns is displayed right-justified. That is, the data is displayed so that the last character of each number is shown at the extreme right of the column.

Show char position range

Indicates that you want ZDT/Db2 to show the start and end character positions for long character columns, in SNGL display mode. The default is not selected.

/

When the data for a character column requires two or more rows on the SNGL display, the second and subsequent rows include a character position range, for example, *ssss-nnnn*. *ssss* is the character position of the first character shown in the row. *nnnn* is the character position of the last character shown in the row.

(blank)

The character position range is not shown.

You can toggle the character position on and off from within the editor using the CHARPOS editor primary command.

Show excluded shadow lines

Controls whether a shadow line, which indicates one or more excluded rows, is displayed. The default is selected.

The ZDT/Db2 editor allows rows to be excluded from display, using the EXCLUDE editor primary command, or the X (and its variations) editor prefix commands. Like the ISPF editor, excluded rows are shown using a shadow line. Each shadow line marks the location of one or more contiguous excluded rows. However, unlike the ISPF editor, the excluded shadow lines can be hidden from display by de-selecting this option.

/

Display excluded shadow lines.

(blank)

Do not display excluded shadow lines.

You can change the display status of excluded rows, from within the ZDT/Db2 editor, by issuing the SHADOW X OFF or SHADOW X ON commands.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (3 of 8) panel, you then press the NxtPage function key (F11) twice.

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (3 of 8) panel if the **Table/Single format, prefix area and shadow options** option on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel...	Use/do this
Editor Options (2 of 8) panel on page 649	Press the PrvPage function key (F10)
Editor Options (4 of 8) panel on page 662	Press the NxtPage function key (F11)

Related concepts

- TABL display format on page 153
- SNGL display format on page 156

Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126
- Selecting a display format on page 153
- Managing the prefix area on page 165
- Showing or hiding excluded rows on page 164

Editor Options (4 of 8) panel

You use the Editor Options (4 of 8) panel to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)             Editor Options (4 of 8)             Global Settings

Primary and Foreign Keys, Indexes:
   Enter "/" to select option
   /  Allow updates to primary key
   /  Show primary key
   /  Show foreign key
   /  Show index indicators
   /  Show nullable column indicators

Export Command:
   Enter "/" to select option
   _  Show export options







Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F12=Cancel
```

Allow updates to primary key

Indicates if you can edit the primary key information. The default is selected.

/

Changes to primary key information allowed.

(blank)

Changes to primary key information not allowed.

Show primary key

Indicates if you want ZDT/Db2 to display primary key information. The default is selected.

/

Include information showing the columns that comprise the primary key for the table.

In table display format, ZDT/Db2 shows the primary key information only if you have also specified that the scale line is to be displayed (see **Show scale line** in Editor Options (3 of 8) panel on page 655). The scale line shows a *"P"* in the first position of each column that forms part of the table's primary key.

In single display format, if the **Show keys, indexes** option is selected (see **Show keys, indexes** in Editor Options (3 of 8) panel on page 655), ZDT/Db2 displays an additional column, immediately to the left of the **Column name** column (and immediately to the right of the **Column number** column, if displayed). This column is 4 characters wide. The presence of a *"P"* in the first position of the column indicates that the column is part of the table's primary key.

(blank)

Do not include information showing the columns that comprise the primary key for the table.

Show foreign key

Indicates if you want ZDT/Db2 to display foreign key information. The default is selected.

/

Include information showing the columns that are part of a foreign key.

In table display format, ZDT/Db2 shows the foreign key information only if you have also specified that the scale line is to be displayed (see Editor Options (3 of 8) panel on page 655). The scale line shows an *"F"* in the fourth position of each column that forms part of a foreign key.

In single display format, if the **Show keys, indexes** option is selected (see Editor Options (3 of 8) panel on page 655), ZDT/Db2 displays an additional column, immediately to the left of the **Column name** column (and immediately to the right of the **Column number** column, if displayed). This column is 4 characters wide. The presence of an *"F"* in the fourth position of the column indicates that the column is part of a foreign key.

(blank)

Do not include information showing the columns that are part of a foreign key.

Note:

1. For tables with multiple foreign keys defined, the indicator character cannot be used to distinguish between columns that are part of the same foreign key or part of different foreign keys.

2. The foreign key information is only displayed if it is stored in the template. If the **Retrieve foreign key information when building templates** system option was not selected when the template for the object was built, no foreign key indicators are displayed.

Show index indicators

Indicates if you want ZDT/Db2 to display index information. The default is selected.

/

Include information showing the columns that are part of an index.

The INCLUDE option of the CREATE INDEX command can be used to append columns to the set of index key columns of a unique index. Such columns form part of the unique index but are not part of the unique index key.

In table display format, ZDT/Db2 shows the index

information only if you have also specified that the scale line is to be displayed (see Editor Options (3 of 8) panel on page 655).

- In the second position of the scale line, a *"U"* is displayed for each column that forms part of a unique index and part of the unique index key.
- In the second position of the scale line, an *"I"* is displayed for each column that forms part of a unique index, but is not part of the unique key.
- In the third position of the scale line, an *"N"* is displayed for each column that forms part of a non-unique index.

See Table Edit panel on page 868.

In single display format, if the **Show keys, indexes** option is selected (see Editor Options (3 of 8) panel on page 655), ZDT/Db2 displays an additional column, immediately to the left of the

Column name column (and immediately to the right of the **Column number** column, if displayed). This column is 4 characters wide.

- In the second position of this column, a *"U"* is displayed for each column that forms part of a unique index and part of the unique index key.
- In the second position of this column, an *"I"* is displayed for each column that forms part of a unique index, but is not part of the unique key.
- In the third position of this column, an *"N"* is displayed for each column that forms part of a non-unique index.

(blank)

Do not include information showing the columns that are part of an index.

Note: For tables with multiple foreign keys defined, the indicator character cannot be used to distinguish between columns that are part of the same index or part of different indexes.

Show nullable column indicators

Indicates if you want ZDT/Db2 to display nullable status for each column. The default is not selected.

/

Include information showing the columns can contain NULL values.

In table display format, ZDT/Db2 shows the nullable column information only if you have also specified that the scale line is to be displayed (see Editor Options (3 of 8) panel on page 655).

In the first position of the scale line, an **n** is displayed for each column that allows NULL values.

(blank)

Do not include information showing the columns are nullable.

Note: When the column is part of the primary key of the table, the first position will contain a **P** showing the column belongs to PRIMARY KEY. In this situation, the column will always be not nullable.

Show export options

The EXPORT primary command can be issued in a Z Data Tools/Db2 editor session to export data to a external file. Issuing the EXPORT command can optionally display the first of the export options panels, where you can specify the data format for the export operation. The default is not selected.

/

The EXPORT primary command dialog displays the first of the export options panels.

(blank)

The EXPORT primary command dialog does not display the first of the export options panels.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (4 of 8) panel, you then press the NxtPage function key (F11) three times.

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (4 of 8) panel if the **Key, index and export options** option on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel…	Use/do this
Editor Options (3 of 8) panel on page 655	Press the PrvPage function key (F10)
Editor Options (5 of 8) panel on page 667	Press the NxtPage function key (F11)

Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126
- Editing related tables on page 192
- Copying data to a VSAM or QSAM file on page 279

Editor Options (5 of 8) panel

You use the Editor Options (5 of 8) panel to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)             Editor Options (5 of 8)             Global Settings

Varying Length Columns:
   Enter "/" to select option         String Delimiters:
  /  Disable input delimiter            Input . . . . . . . #
  /  Show end of string                 Display . . . . . . <
  /  Remove trailing spaces

Columns that allow Null Values:
   Enter "/" to select option         Null Column Indicators:
   _  Convert spaces to Db2 null         Input . . . . . . . @
                                         Display . . . . . . _

Zoned Timestamp Columns:
   Enter "/" to select option
   /  Auto-size display width



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F12=Cancel
```

Disable input delimiter

This option is used to disable processing for the Input string delimiter. By default this option is not selected. When a change is made to a varying field, ZDT/Db2 scans the entire varying field from left to right, looking for the first occurrence of the Input string delimiter character. If an occurrence is found, the field is truncated at the character before the location of the Input string delimiter. The default is not selected.

Consider disabling the Input string delimiter when:

- The Db2® object being processed contains VARCHAR fields that cannot be displayed in full, without scrolling. In this situation a change to the field may trigger silent truncation when the input string delimiter is present in a part of the field that is not currently visible on the display.
- A VARCHAR field is used to store data other than readable text, for example binary data, or data that has an internal format. In this situation the data may contain a character that matches the input string delimiter character, leading to unwanted truncation if the data in that field is changed.

Show end of string

This option only applies to the processing of varying-length (VARCHAR) input fields. The default is selected. Specifies if the end of a varying-length string is to be indicated using the delimiter specified in the **String delimiters: Display** option.

/

Show the end of a varying-length string using the delimiter specified in the **String delimiters: Display** option. If the string is of maximum length, the delimiter character is not shown.

(blank)

Do not show the end of a varying-length string.

Remove trailing spaces

This option only applies to the processing of varying-length (VARCHAR) input fields. The default is selected. When you enter a value that is less than the maximum length for the column, determines if ZDT/Db2 removes trailing spaces:

/

Remove trailing spaces and store the specified value using the minimum length string.

(blank)

Do not remove trailing spaces.

String Delimiters

These options control the end of string delimiter when entering and displaying data in a varying column.

Input

This option only applies to the processing of varying-length (VARCHAR) input fields.

For input, ZDT/Db2 sets the end-of-string delimiter character to this value. If you enter data that requires one or more trailing spaces in a varying-length field, you can use this option to provide a character of your choice to indicate the end of the string. The default value is the number sign (#).

Valid values: any character.

Display

This option only applies to the processing of varying-length (VARCHAR) input fields. ZDT/Db2 sets the end-of-string delimiter character, for display, to a this value. If you have also selected **Show end of string**, ZDT/Db2 displays the value in the field, terminated with the delimiter character you have selected. If the column is of maximum length, the delimiter character is not shown. The default value is the *"less than"* sign (<).

Valid values: any character.

Convert spaces to Db2® null

When any column that allows null values is changed so that it contains only spaces, the value of the column is set to null.

If the column is defined as NOT NULL, the column remains unchanged. The default is not selected.

Null Column Indicators

These options allow the specification of characters to represent the Db2® null value when entering and displaying data.

Input

A single character which, when entered in the first byte of an input field, causes ZDT/Db2 to update the value for the column to NULL. The default value is the *"at"* symbol (@).

Valid values: any character

If the column allows null values, when you enter this character in the first byte of the input field, ZDT/Db2 updates the value for the column to NULL. Any non-blank character following the indicator invalidates the action.

If the column is defined as NOT NULL, ZDT/Db2 treats the character in the same way as any other input character.

Display

A single character indicating that the column contains a null value (as opposed to, for example, blanks). The default value is the underscore character "_".

Valid values: any character

Auto-size display width

An option that modifies the default display width for applicable columns in TABL display mode. The option applies to zoned timestamp columns only. The default is selected.

This option is only applicable when a Db2® system that supports zoned timestamp data types is connected. Zoned timestamps were first included with Db2® version 10.

/

Allows ZDT/Db2 to automatically adjust the display width for zoned timestamp columns to that needed to display the data.

(blank)

ZDT/Db2 shows zoned timestamp columns with a field width in the range 147 - 160 characters. Most of the data displayed will be blank, because the maximum required display width for a zoned timestamp excluding the use of exits and field procs is 39 characters.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (5 of 8) panel, you then press the NxtPage function key (F11) four times.

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (5 of 8) panel if the **Varying length and nullable column options** option on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel…	Use/do this
Editor Options (4 of 8) panel on page 662	Press the PrvPage function key (F10)
Editor Options (6 of 8) panel on page 672	Press the NxtPage function key (F11)

Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126
- Varying-length columns on page 178
- Null values and VARCHAR processing on page 178

Editor Options (6 of 8) panel

You use the Editor Options (6 of 8) panel to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)             Editor Options (6 of 8)             Global Settings

Enter Key Processing:
   Enter key usage
   1  1. No action
      2. Save data
      3. Save data and commit changes

Commit Options:
   Enter "/" to select option
   /  Commit when save issued
   _  Commit when no save errors

Auto Commit (Changes):
   Auto-commit count . . . . . .  0_________



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F12=Cancel
```

Enter key usage

Set this option to one of the following:

1. No action

This is the default setting. No special action is taken when the Enter key is pressed and no other operations are pending. Any changes made during an Edit session are only presented to Db2® when an explicit SAVE command is issued, or at the end of the Edit session.

2. Save data

All pending changes are submitted to Db2® for processing whenever the Enter key is pressed (and no other operations are pending). Successful changes are not committed until the end of the Edit session, unless the **Commit changes when Save issued** option is selected. Rows for which

there was an unsuccessful change are marked with *"=ERR"*, *"ERR R"*, or the SQLCODE in the prefix area (see Show SQLCODE on page 658).

3. **Save data and commit changes**

   All pending changes are submitted to Db2® for processing whenever the Enter key is pressed (and no other operations are pending). Rows for which there was an unsuccessful change are marked with *"=ERR"*, *"ERR R"*, or the SQLCODE in the prefix area (see Show SQLCODE on page 658). ZDT/Db2 issues an explicit Db2® commit at the end of the save process; however, this only occurs if:

   - No save errors were encountered, or
   - **Commit when no save errors** is turned off.

   Committing the changes to Db2® is an irrevocable process.

Commit when save issued

The default for this option is selected. The setting for this option may have been fixed by the product installer. ZDT/Db2 issues an explicit Db2® commit at the end of the save process; however, this only occurs if:

- No save errors were encountered, or
- **Commit when no save errors** is turned off.

Note that the Enter key processing options **Save data** and **Save data and commit changes** behave as though an explicit save command had been issued when the Enter key was pressed.

Commit when no save errors

The default for this option is not selected. The setting for this option may have been fixed by the product installer.

When this option is not selected, ZDT/Db2 issues an explicit Db2® commit at the end of the save process, regardless of whether any errors were encountered during the save process.

When this option is selected, ZDT/Db2 only issues an explicit Db2® commit at the end of the save process when no errors were encountered during the save process.

Auto-commit count

The default action, indicated by a zero value, is that no automatic Db2® commit points are during the editor SAVE process. When a positive value is specified, ZDT/Db2 issues an automatic Db2® commit point whenever there have been *value* successful changes during the SAVE process (where *value* is the value specified for **Auto-commit count**). Specifying a non-zero value can cause multiple Db2® commit points to be issued during the SAVE process.

Example:

With an auto-commit count of 2 and 11 successful changes processed during a SAVE process, five Db2® commit points are issued. The changes for the last row processed remain uncommitted.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (6 of 6) panel, you then press the NxtPage function key (F11) five times.

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (6 of 8) panel if the **Enter key and commit processing options** option on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel…	Use/do this
Editor Options (5 of 8) panel on page 667	Press the PrvPage function key (F10)
Editor Options (7 of 8) panel on page 675	Press the NxtPage function key (F11)

Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126
- Ending a Z Data Tools/Db2 editor session on page 133

Editor Options (7 of 8) panel

You use the Editor Options (7 of 8) panel to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)             Editor Options (7 of 8)             Global Settings

Concurrency Options:                  SQL Cursor Concurrency Options:
   Enter "/" to select option            Concurrency
   _  Read-only access                   1  0. Uncommitted read
   /  Commit after data fetch               1. No with clause
   _  Skip locked rows                      2. Cursor stability
                                            3. Read stability
Edit Locking Option:                        4. Read stability keep locks
   Locking                                  5. Repeatable read
   1  1. None                               6. Repeatable read keep locks
      2. Share mode
      3. Exclusive mode                  Keep locks option
                                         1  1. Use/Keep share locks
                                            2. Use/Keep update locks
                                            3. Use/Keep exclusive locks



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F12=Cancel
```

Concurrency options

For a detailed explanation about this subject, refer to the Db2 UDB for z/OS Administration Guide.

Concurrency options affect the way Db2® takes row and page locks for data in a Db2® table. Locking options affect the way Db2® takes table locks. When a table locking option is specified, it overrides any concurrency option.

In general terms, specifying a higher value for the concurrency option reduces the likelihood of deadlocks when editing an object, at the expense of other users' ability to access and change data.

ZDT/Db2 provides options that can minimize Db2® locking, and the duration of the Db2® locks taken, when ZDT/Db2 accesses Db2® and retrieves data for a Z Data Tools/Db2 editor session.

ZDT/Db2 also provides options to lock the table to prevent other users from either updating or reading data. The locking options should be used with caution.

Note that both the concurrency and table locking options only apply to the initial fetch of data from Db2®. If either of the commit options **Commit changes when save issued** or **Commit changes when no save errors** is specified, each commit point releases any locks held by Db2® against the table. For more information, see the description about **Enter Key Processing, Commit Options** in Editor Options (6 of 8) panel on page 672.

After each commit point, ZDT/Db2 re-locks the table if a table-locking option has been specified; however, other Db2® users may access the table between the time of the commit point and ZDT/Db2 re-locking the table.

Note:

1. Use table locking options with caution.
2. For the table locking option to be effective, the Db2® object must be a table. It is not possible to lock a view.
3. Any concurrency option applies to the initial fetch of data from Db2® only.
4. Any concurrency or locking option persists only until the next commit point. See below for more information.
5. Set the **Concurrency** option to 0.

ZDT/Db2 provides options for accessing data in *"read-only"* mode, and for reducing the life of the Db2® locks taken when data is retrieved from Db2®. It also provides control over the locking mechanism used when data is retrieved from Db2®.

Read-only access

If you select this option, ZDT/Db2 accesses data from Db2® in *"read-only"* mode, by adding the FOR FETCH ONLY clause to the SQL statement used to retrieve data from Db2®. By default, this option is not selected.

Selecting this option does not prevent edit operations from occurring.

Note that, for a Z Data Tools/Db2 Browse session, the FOR FETCH ONLY clause is always added to the SQL statement used to retrieve data from Db2®.

Commit after data fetch

The default for this option is selected. The value for this option may have been fixed by the product installer. If you select this option, ZDT/Db2 issues an explicit COMMIT when the data from the ZDT/Db2 Edit session was retrieved from Db2®. This releases all Db2® locks taken when the data for the ZDT/Db2 Edit session is retrieving. It also releases any explicit table locks taken when a locking shared or exclusive option is specified.

This option is recommended in high concurrency environments, where it is important to minimize the duration of any Db2® locks held.

Selecting this option does not prevent edit operations from occurring; however, it increases the likelihood of errors when changed data is presented to Db2® for validation. This is because other processes are free to change the Db2® data for which a copy is stored in the ZDT/Db2 Edit session. Therefore, when a change made in the ZDT/Db2 Edit session is presented to Db2®, the original data might have changed, resulting in various SQL errors.

Note that, for a Z Data Tools/Db2 Browse or View session, ZDT/Db2 always issues an EXPLICIT commit once the data for the Browse session has been retrieved from Db2®.

Skip locked rows

Use this option to control whether a searched DELETE statement ignores or skips any rows that are incompatibly locked by other processes. The default is not selected.

Edit Locking Option

When ZDT/Db2 operates in edit mode only, use this option to control whether ZDT/Db2 explicitly locks the Db2® object that is being processed. This option applies to tables only, not to objects that are defined on a table such as views. This option may have been fixed by the product installer. When the option is available for update, the default is 1 (**None**).

1. None

ZDT/Db2 does not issue an explicit lock before accessing data for the table. This is the default choice.

2

ZDT/Db2 issues a statement that locks the table in share mode before accessing data for the table. The data from the table can be read by other tasks. However, the table cannot be updated by other processes until the ZDT/Db2 edit session ends. Use this option with caution to guarantee access to the table.

3

ZDT/Db2 issues a statement that locks the table in exclusive mode before accessing data for the table. The data from the table cannot be either read or written by other processes until the ZDT/Db2 edit session ends. Use this option with extreme caution to guarantee access to the table.

SQL Cursor Concurrency Options (Concurrency)

ZDT/Db2 can add a WITH clause to the SQL statement used to fetch data from Db2®. When a WITH clause is specified, Db2® takes locks (of varying levels) against either rows or pages within the table, as the data is accessed. The following options are possible. The default is 1 (**No with clause**).

0. Uncommitted read

Db2® locking is disabled. The data retrieved from Db2® may be inconsistent if another process is rolled back after the data access. Use this option in a high concurrency environment.

1. **No with clause**

   This is the default value and recommended for most application programs. ZDT/Db2 does not add a WITH clause to the SELECT statement for the table. Any rows retrieved by ZDT/Db2 initially have a S(hare) lock. With this option it is possible that other processes can update the table you are editing.

2. **Cursor stability**

   A row or page lock is held only long enough for the cursor to move to another row or page.

3. **Read stability**

   A row or page lock is held for rows or pages that are read by the editor at least until the next commit point.

4. **Read stability keep locks**

   A row or page lock is held for rows or pages that are read by the editor at least until the next commit point. The **Read stability** option takes an *"S"* or *"U"* lock, while this option takes an *"X"* lock on affected rows or pages.

5. **Repeatable read**

A row or page lock is held for all accessed rows, qualifying or not, at least until the next commit point. ZDT/Db2 can return to a previously read page and read the same row again, and the data is not changed and no new rows are inserted.

6. Repeatable read keep locks

This option is similar to the **Repeatable read** option, except that ZDT/Db2 takes an *"X"* lock, rather than a *"S"* or *"U"* lock.

SQL Cursor Concurrency Options (Keep locks option)

ZDT/Db2 can add a WITH clause to the SQL statement used to fetch data from Db2®. When a WITH clause is specified, Db2® takes locks (of varying levels) against either rows or pages within the table, as the data is accessed. The following options are possible. The default is 1 (**Use/Keep share locks**).

1. Use/Keep share locks

ZDT/Db2 does not issue an explicit table lock for the object being edited. Locking of rows and pages within the object is managed by Db2® according to ZDT/Db2 concurrency options and Db2® installation and object definition options.

2. Use/Keep update locks

ZDT/Db2 issues a LOCK TABLE IN SHARE MODE statement before reading the data from Db2®. This restricts all other users to read-only operations on the object. If either the Enter key option **Save data and commit changes**, or the **Commit changes when SAVE issued** option is selected, any commit point take during the Edit session releases the table lock. ZDT/Db2 re-issues the lock immediately after the commit point. However, other processes can access the table between the time of the commit point and ZDT/Db2 re-locking the table.

3. Use/Keep exclusive locks

ZDT/Db2 issues a LOCK TABLE IN EXCLUSIVE MODE statement before reading the data from Db2®. This prevents all other users from accessing the object, unless the other process is running with an isolation level of uncommitted read. If either the Enter key option **Save data and commit changes**, or the **Commit changes when SAVE issued** option is selected, any commit point take during the Edit session releases the table lock. ZDT/Db2 re-issues the lock immediately after the commit point. However, other processes can access the table between the time of the commit point and ZDT/Db2 re-locking the table.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (7 of 8) panel, you then press the NxtPage function key (F11) six times.

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (7 of 8) panel if the **Concurrency and locking options** option on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel…	Use/do this
Editor Options (6 of 8) panel on page 672	Press the PrvPage function key (F10)

Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126

Editor Options (8 of 8) panel

You use the Editor Options (8 of 8) panel to set your default editor options.

Note: See the description for the Editor Options (1 of 8) panel (Editor Options (1 of 8) panel on page 644) about how long any changes you make to these options last.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)             Editor Options (8 of 8)             Global Settings

LOB and XML Columns:
   Enter "/" to select option
   _  Show part/all data              Max characters  . . 0___

LOB Display Format:                   LOB Piece Size:
   Initial display format                Split LOBs into pieces using
   3  1. Previous                        2  1. Table display size
      2. Table                             2. Single display size
      3. Single                            3. Manual ... 100__


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F12=Cancel
```

Show part/all data

When the Db2® object contains LOB (BLOB, CLOB or DBCLOB), or XML columns, you can display some, or possibly all of the data within those columns. The default action is not selected, in which case no data is displayed. The LOB and XML column appears in the editor as a column heading only, with no data displayed.

To display some, or possibly all, of the data in a LOB or XML column, select the option. You must also specify a maximum number of characters to displayed using the Max characters option.

Note that you can display all of the data in a LOB or XML column using the LOBBRWS, LOBVIEW, LOBEDIT, XMLBRWS, XMLVIEW and XMLEDIT editor primary commands.

Max characters

This option works in conjunction with the Show part/all data option. You can specify a value in the range 0-1000. The default is 0. When the Show part/all data option is selected, and a non-zero value is specified for the Max characters option, ZDT/Db2 retrieves and displays that many characters of the LOB or XML data in the editor session.

Initial Display Format

Use this section of the Editor Options panel to specify how editor panels appear at the start of a Z Data Tools/Db2 LOB editor session. This option works with the Split LOBs into pieces using option. The default is 3 (**Single**). Set this option to one of the following:

1. **Previous**

   The display format field (Format) on the LOB editor panel is pre-filled with the value specified in the previous ZDT/Db2 LOB editor session.

2. Table

The LOB Editor panel is displayed in TABLE display format. When this option is selected it is suggested that the Split LOBs into pieces using option be set to *"Table display size"*.

3. Single

The LOB Editor panel is displayed in SNGL display format. This is the default setting. When this option is selected it is suggested that the Split LOBs into pieces using option be set to *"Single display size"*.

Split LOBs into pieces using

This option determines the size of the pieces, when ZDT/Db2 displays a LOB column in the LOB editor. LOB data can be very large, and a 3270 display is not well suited to displaying large columns directly, so the LOB editor breaks the LOB column into pieces, and displays those pieces as though the LOB column was actually a table with a single varying column.

The length of the varying column is the maximum length of any piece. The default is 2 (**Single display size**). This option works in conjunction with the Initial Display Format option (for LOBs) as follows:

1. Table Display size

The size of each piece is approximately the same as the number of characters displayed on a single line in TABL display format. The actual number of characters displayed depends on the terminal characteristics. This may not be the best choice when processing large objects because of the large number of pieces required.

2. Single Display size

The size of each piece is approximately the same as the number of characters displayed on the terminal in SNGL display format. The actual number of characters displayed depends on the terminal characteristics. On a 24x80 terminal the value is approximately 850; on a 48x132 terminal the value is approximately 4250.

This allows large amounts of data to be displayed at one time – when compared with the Table Display size option.

Usage tip 1: Turn on the character position range indicator by issuing the CHARPOS command. This shows the starting and ending character position for each row of data on the display.

Usage tip 2: You can use the NEXT (PF11 by default) and PREVIOUS (PF10 by default) PFKeys to scroll through the LOB data. The character position range indicator values show the absolute position from the start of the LOB, rather than the relative position within each piece of the LOB.

3. Manual

You can specify your preference for the piece size by specifying '3'.

Enter your preference for piece size in the Manual input field. A value in the range 60-32760 is required. The default is 100.

Parent panels

You can go to the Editor Options (1 of 8) panel from the *"parent"* panels shown below. To go to the Editor Options (8 of 8) panel, you then press the NxtPage function key (F11) seven times.

- Set Processing Options panel on page 832
- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Action bar pull-down menu on page 445

You can only display the Editor Options (8 of 8) panel if the **XML and LOB column processing** options on the Editor Options (1 of 8) panel is selected.

Child panels

To display this panel…	Use/do this
Editor Options (7 of 8) panel on page 675	Press the PrvPage function key (F10)

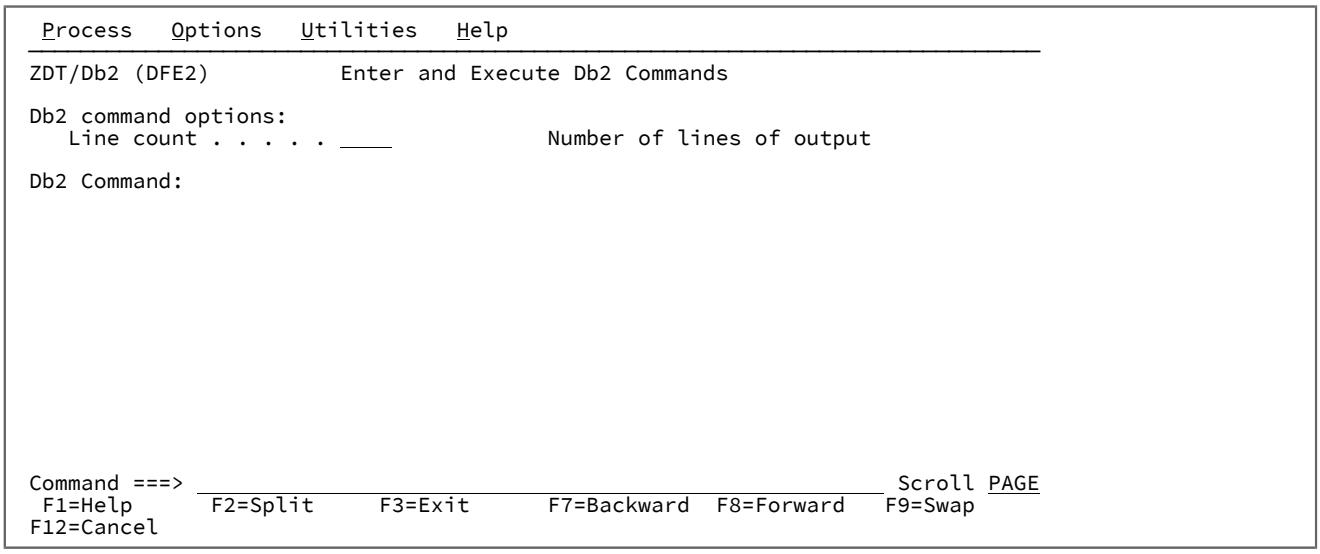
Related tasks

- View and Edit options (options 1 and 2) on page 67
- Setting default processing options on page 62
- Setting options for the current ZDT/Db2 editor session on page 126

Enter and Execute Db2® Commands panel

You use the Enter and Execute Db2® Commands panel to enter Db2® commands and view the output returned by Db2®.

Panel and field definitions



Line count

Only applies to DISPLAY THREAD and DISPLAY DATABASE commands. This figure controls the number of lines to be returned and is included in the LIMIT parameter for the command which has been entered.

Db2® Command

The Db2® command to be issued. This command may or may not be prefixed by a hyphen (-). If it is not prefixed by a "-", a "-" is added.

Parent panels

- Primary Option Menu panel on page 760
- Any panel where the Process pull-down menu is available.

Child panels

None.

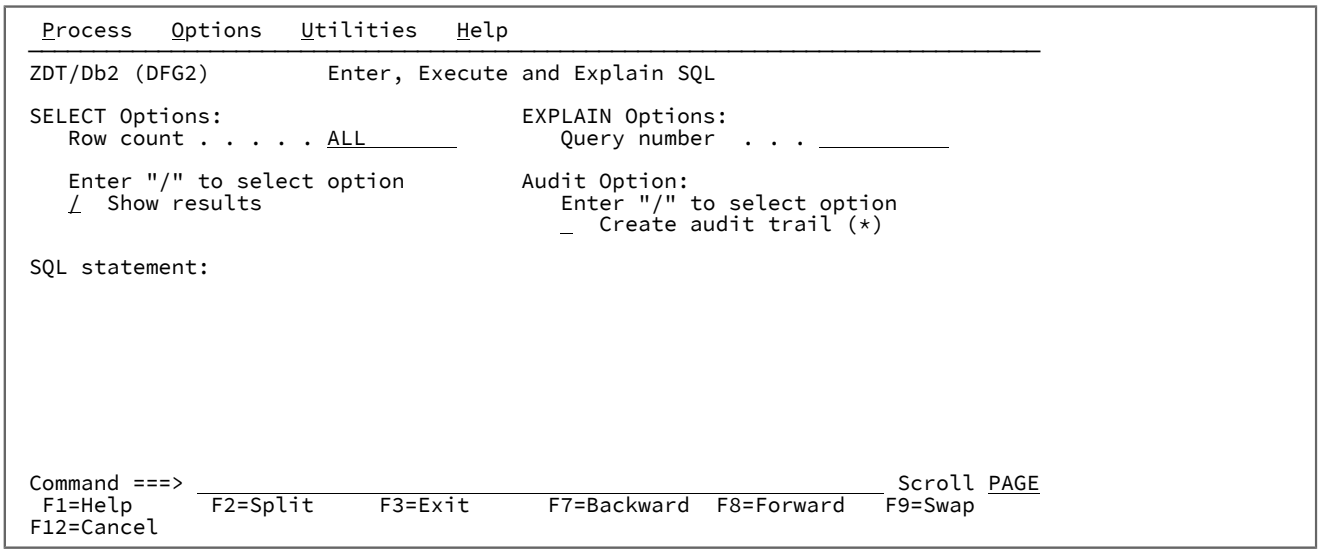
Related tasks

- Issuing Db2 commands from within a Z Data Tools/Db2 session on page 200

Enter, Execute and Explain SQL Statements panel

You use the Enter, Execute and Explain SQL Statements panel to enter freeform SQL statements, run freeform SQL statements, or obtain information about an SQL statement.

Panel and field definitions



Row count

Limits the rows loaded into the ZDT/Db2 editor when the SQL statement being processed is a SELECT statement. To load all rows for the SELECT statement, specify "*" or ALL in this field.

Query number

A number to help identify the results of this explanation. If you leave this field blank, ZDT/Db2 generates a query number for you in the format YYYYMMDD*ss*, where *ss* is a sequence number.

Show results

If selected, ZDT/Db2 attempts to display the relevant rows that Db2® has inserted in the plan table as a result of explaining the statement.

Create audit trail

Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

SQL statement

Freeform SQL statement entry area. If you want to enter more than one SQL statement, you must separate the statements with a semicolon (;).

Parent panels

- SQL Prototyping, Execution and Analysis panel on page 843

Child panels

None.

Related tasks

- Entering SQL statements on page 425
- Executing SQL statements on page 426
- Explaining SQL on page 427
- SAF-rule controlled auditing on page 71

Related references

- DOWN primary command on page 939
- EXECUTE primary command on page 948
- EXPLAIN primary command on page 950
- SQL primary command on page 1005
- UP primary command on page 1009

EXPLAIN Interpretation panel

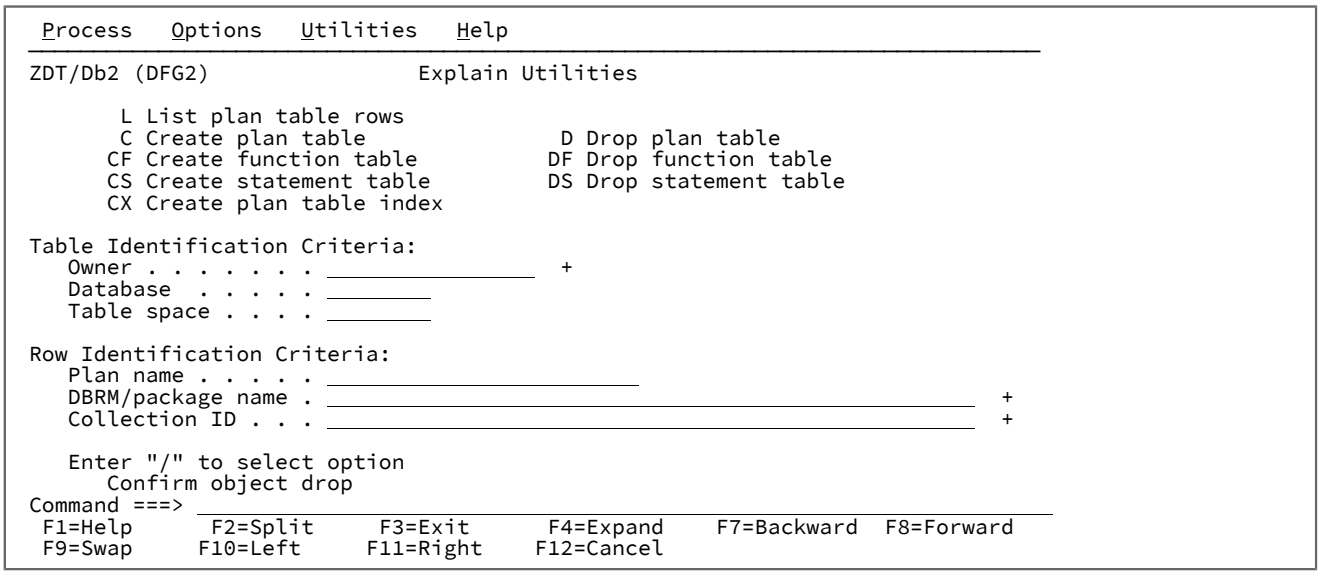
ZDT/Db2 displays the EXPLAIN Interpretation panel if you use the I line command against an item listed on:

- The Plan Table Rows panel on page 754, or
- The Statement Table Rows panel on page 845

Explain Utilities panel

You use the Explain Utilities panel to select functions to support the use of the Db2® EXPLAIN SQL statement.

Panel and field definitions



L

List a plan table. ZDT/Db2 displays the rows of the specified plan table. If you leave the **Plan table owner** field blank, ZDT/Db2 uses your current SQL ID. You can qualify the rows by **Plan name**, **DBRM/package name**, and **Collection ID**. You can list any related statement table and function table rows, if they exist, using line commands from the plan table. The table name for a plan table is always PLAN_TABLE.

C

Create a plan table. A plan table is needed for the SQL EXPLAIN statement. If you leave the **Plan table owner** field blank, ZDT/Db2 uses your current SQL ID. You can optionally use the **Database** and **Table space** fields to define the database and table space to be used when the plan table is created. The table name for a plan table is always PLAN_TABLE.

CF

Create a function table, where the SQL EXPLAIN statement can store information about how function references are resolved. ZDT/Db2 uses the **Plan table owner** field (or, if it is left blank, your current SQL ID) as the owner of the function table. The table name for a function table is always DSN_FUNCTION_TABLE.

CS

Create a statement table, where the SQL EXPLAIN statement can store the estimated costs for an SQL statement. ZDT/Db2 uses the **Plan table owner** field (or, if it is left blank, your current SQL ID) as the owner of the statement table. The table name for a statement table is always DSN_STATEMNT_TABLE.

CX

Create an index on the plan table for the Db2® optimizer. An index on the plan table is recommended if you use Db2® optimizer hints.

U

Upgrade a plan table to the current Db2® version. Use this option to add columns so that the plan table is at the level that Db2® supports.

D

Drop plan table.

DF

Drop function table.

DS

Drop statement table.

Plan table owner

See function descriptions below.

Database

See function descriptions below.

Table space

See function descriptions below.

Plan name

See function descriptions below.

DBRM/package name

See function descriptions below.

Collection ID

See function descriptions below.

Confirm object drop

Indicates if you want ZDT/Db2 to display a confirmation panel each time you enter a DR(op) command against an object

Parent panels

- SQL Prototyping, Execution and Analysis panel on page 843

Child panels

To display this panel…	Use/do this
Plan Table Rows panel on page 754	Function L
Create Index panel on page 542	Function CX

Related tasks

- Managing tables used by SQL explain on page 430

Export Options (1 of 3) panel

You use the Export Options (1 of 3) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 120. Export Options (1 of 3) panel

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)            Export Options - (1 of 3)           Global Settings

Data Format:
   1  1. ZDT/Db2 (SQLDA) format       Enter "/" to select option
      2. Db2 UNLOAD format            /  Native unicode processing
      3. DSNTIAUL format
      4. User defined
      5. Delimited variables (CSV)

Execution options:
   Select option
   2  1. Online                       Enter "/" to select option
      2. Batch                  -->   _  Batch data set creation
      3. Batch, using Db2 UNLOAD -->     Edit Db2 UNLOAD options

More Options:
   Enter "/" to select option
   /  Null indicators / CSV options   (For user-defined, CSV data formats)
      Data type format                (For user-defined data format)
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F11=NxtPage  F12=Cancel
```

Data format

The format in which Export Utility unloads data:

- ZDT/Db2 (SQLDA) format. Db2® internal format. This is the default.
- Db2® external (or unload) format.
- Db2® DSNTIAUL unload format. This is the format used by the Db2® DSNTIAUL sample program.
- User-defined format.
- Delimited Variables (CSV) format.

  ZDT/Db2 writes the data to the output file as a comma-delimited (CSV) file.

The data for each column appears in the output record in column number order, and each data value is separated from the next by a separator character.

Db2® data exported in this way cannot be imported using Import (option 3.6). The data is intended for use by a spreadsheet application program.

Native unicode processing

Determines how ZDT/Db2 processes Unicode encoded data when exporting data from a table to a dataset. Normally data stored in Db2® in Unicode is converted automatically to the CCSID of the Db2® plan, typically CCSID 37. This automatic translation results in conversion errors when a Unicode character does not have an equivalent code point in CCSID 37. When data is exported in "native Unicode" format, the automatic translation of data from unicode to the CCSID of the Db2® plan is suppressed. This ensures that there are no data conversion errors, but means the data is not directly visible using z/OS® utilities such as the editor. Use this option if the data is to be transferred to other systems that support Unicode data. When the option is not enabled, the option has no effect; that is, the default processing is used. When the data being exported is not Unicode encoded, the option has no effect, even if enabled.

/

Process Unicode data natively.

(blank)

Do not Process Unicode data natively. This is the default setting.

Execution Options

Online

The export is run in foreground.

Batch

Builds the necessary JCL to perform the Export utility in a batch job, and displays it in an edit session so that you can review the JCL prior to submitting the job.

Batch, using Db2® utility

Selecting this option runs the export using the Db2® UNLOAD utility.

Generates the JCL for the UNLOAD job, and displays it in an edit session so that you can review the JCL prior to submitting the job.

The default is **Batch**.

Batch data set creation

This option only applies when batch execution is selected, and is ignored for online execution. You use this option to defer the creation of the export data set to batch job execution time. When this option is not selected, ZDT/Db2 determines whether the export data set exists before generation of the batch job JCL. If the data set does not exist, you are prompted to create it.

The default is not selected, which means that export data set creation is not deferred to batch job execution time. When you set the execution option to **Batch, using Db2® utility**, this option is automatically selected and cannot be changed.

Edit Db2® UNLOAD options

Used in conjunction with option **3 Batch, using Db2® UNLOAD**. The UNLOAD Options panel is displayed after you press the Exit function key (F3). The default is not selected.

More Options

Null indicators / CSV options

If selected, you can navigate to the Export Options (2 of 3) panel by pressing the NxtPage function key (F11).

This option is only selectable if you have also selected either the data format option **4. User defined** or **5. Delimited variables (CSV)**

When this option is displayed, the default is not selected.

.

Data type format

If selected, you can navigate to the Export Options (3 of 3) panel by pressing the NxtPage function key (F11).

This option is only selectable if you have also selected the data format option **4. User defined**.

When this option is displayed, the default is not selected.

Parent panels

- Set Utility Options panel on page 840

Child panels

To display this panel...	Use/do this
Export Options (2 of 3) panel on page 692	Select the **Null indicators / CSV options** option (data format option 4 or 5 must also be selected), and press the NxtPage function key (F11).

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- Copying data to a VSAM or QSAM file on page 279

Export Options (2 of 3) panel

You use the Export Options (2 of 3) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 121. Export Options (2 of 3) panel

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)            Export Options - (2 of 3)            Global Settings

Null Indicators:
   Usage               Placement           Type                Indicator
   1  1. Separate      _  1. Before        2  1. One byte      _ Char
      2. None             2. After            2. Two byte      -1___ Integer
                          3. User defined

CSV Options
   Enter "/" to select option          Separator character
   _  Include column headers            1  1. Character    -->  ,
   _  "ASIS" encapsulation                  2. Integer    -->  -1
   _  Quoted values
   _  Single quotes                    Null indicator
   _  Leading zeros                     1  1. Character    -->
   /  Include leading blanks                   2. Integer  -->  -1
   _  Include trailing blanks                  3. None     -->


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F11=NxtPage  F12=Cancel
```

Usage

For nullable columns, determines whether fields in the exported data correspond to the null indicator:

Separate

A null indicator is output, taking into account the settings for Placement, Type and Indicator.

None

No fields in the exported data correspond to the null indicator.

The default is 1 (**Separate**).

Placement

Determines where the null indicator is placed:

Before

The null indicator is placed before the data for the column to which it applies.

After

The null indicator is placed after the data for the column to which it applies.

User defined

A 1-byte (character format) or 2-byte (signed integer format) null indicator is placed somewhere in the output record. Placement depends on mapping the null indicator to a corresponding field in the output record.

The default is 1 (**Before**).

Type

Determines the type of null indicator to be used:

One byte

1-byte (character format) null indicator.

Two byte

2-byte (signed integer format) null indicator.

The default is 2 (**Two byte**).

Indicator

Determines the value of the null indicator:

Char

The character used as the 1-byte (character format) null indicator. The default is blank.

Integer

The integer used as the 2-byte (signed integer format) null indicator. The default is an integer value of -1.

CSV Options

The following options are available when the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel.

Include column headers

Select this option to produce comma-delimited column headers as the first export record. The default is not selected.

"ASIS" encapsulation

Determines whether automatic encapsulation of column data is disabled. Select this option when the data has already been encapsulated in a user-defined character, prior to storage in Db2®. The default is not selected.

Quoted values

Select this option to enclose the CSV output field value in either double or single quotes. By default, output fields within a CSV file that contain an embedded delimiter, and fields consisting only of blanks, are enclosed in quotes. For single quotes, additionally select the **Single quotes** option.

Single quotes

Select this option to enclose output field values in single quote instead of double quotes.

Leading zeros

Select this option to format all numeric fields with leading zeros.

Include leading blanks

When the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel, Db2 CHAR or VARCHAR data containing leading blanks include the blanks when written to the exported file. If the Db2 data contains only blanks and no other characters, then choose both "Include leading blanks" and "Include trailing blanks" to cause all of the blanks to be written to the exported file. The default is to include leading blanks.

Include trailing blanks

When the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel, Db2 CHAR or VARCHAR data containing trailing blanks include the blanks when written to the exported file. If the Db2 data contains only blanks and no other characters, then choose both "Include trailing blanks" and "Include leading blanks" to cause all of the blanks to be written to the exported file. The default is to not include trailing blanks.

Separator character

The character used to separate data when the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel. The default separator character is a comma.

Null indicator type

Character

A character used to represent a null value when the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel. If Type 1 is specified, the default character used to represent a null value is underscore.

Integer

An integer used to represent a null value when the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel. If Type 2 is specified, the default value used to represent a null value is -1.

None

When the **Delimited variables (CSV)** data format option is specified on the Export Options (1 of 3) panel, and Type 3 is specified, null values within the Db2 data are represented by an empty field in the exported file.

Parent panels

- Set Utility Options panel on page 840

Child panels

To display this panel…	Use/do this
Export Options (1 of 3) panel on page 689	Press the PrvPage function key (F10)
Export Options (3 of 3) panel on page 696	Press the NxtPage function key (F11) (the data format option **4. User defined**, and the **Data type format** option, on the Export Options (1 of 3) panel must be selected).

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- Copying data to a VSAM or QSAM file on page 279

Export Options (3 of 3) panel

You use the Export Options (3 of 3) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 122. Export Options (3 of 3) panel

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)            Export Options - (3 of 3)           Global Settings

Decimal Data:
   Select option
   1  1. Internal (packed) format
      2. Zoned format
      3. External format

Integer Data:
   Enter "/" to select option
   _  External format

Floating Point Data:
   Enter "/" to select option
   _  External format

Graphic Data (fixed length):
   Enter "/" to select option
   _  Use shift-out/shift-in (SOSI) characters

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F11=NxtPage  F12=Cancel
```

Decimal Data

Internal (packed) format

Any decimal field occupies the minimum number of bytes needed to represent the data as a packed decimal field.

Zoned format

In Zoned format the field occupies the number of bytes equal to the precision.

External format

In External format the field occupies the number of bytes equal to the precision + 2.

The default is 1 (**Internal (packed) format**).

Integer Data

External format

In External format the field occupies 6 bytes for smallint(2), 11 bytes for integer(4), 20 bytes for bigint.

The default is not selected.

Floating Point Data

External format

In External format the field occupies 14 bytes for float(4), 23 bytes for float(8).

The default is not selected.

Graphic Data (fixed length)

Use shift-out/shift-in (SOSI) characters

Use shift-out/shift-in (SOSI) characters to enclose the exported graphic fixed length data.

The default is not selected.

Note: External format is the same format that is used to display data in ZDT/Db2 Editor.

Parent panels

- Set Utility Options panel on page 840

Child panels

To display this panel...	Use/do this
Export Options (2 of 3) panel on page 692	Press the PrvPage function key (F10)

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- Copying data to a VSAM or QSAM file on page 279

Export Utility ("From") panel

You use the Export Utility panel to specify the name of the Db2® object that contains the data to be exported, the (optional) name of a template data set, and processing options.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Export Utility

From Db2 Object:
   Location  . . . . . _________________        Database . . _________ (optional)
   Owner . . . . . . . HFMUSER          +       Table space  _________ (optional)
   Name  . . . . . . . EMP                                                  +

   Export Count  . . . ALL_____       Number of rows to export

From Template:
   Data set name . . . ____________________________________________
   Member  . . . . . . ___________

Processing Options:
   Template usage                     Enter "/", "A"lways to select option
   3  1. Above                        _  Edit options
      2. Previous                     _  Edit template
      3. Generate from table          _  Create audit trail (*)
      4. Generate/Replace
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Export Count

The number of rows to be exported. This indicates the number of rows to be read from the Db2® table. It can be either a number in the range 1–99999999, or *, or ALL. The default is ALL, where all rows selected from the Db2® object are exported.

Data set name

The name of the data set or PDS member containing the template for the *"From"* table. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The Export Utility *"From"* panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **From Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (see Template usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **From Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. **Generate/Replace**

   Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **From Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a / or an "A":

Edit options

Select this option to display the first of the Export Options panels. The default is not selected.

Edit template

Select this option to edit the template before exporting the data. The default is not selected.

Create audit trail

Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Parent panels

- Utility Functions panel on page 915

Child panels

To display this panel…	Use/do this
Export Utility ("To") panel on page 701	Select **Edit template**
Column Selection/Edit panel on page 497	Do not select **Edit template**
Export Options (1 of 3) panel on page 689	Select **Edit options**

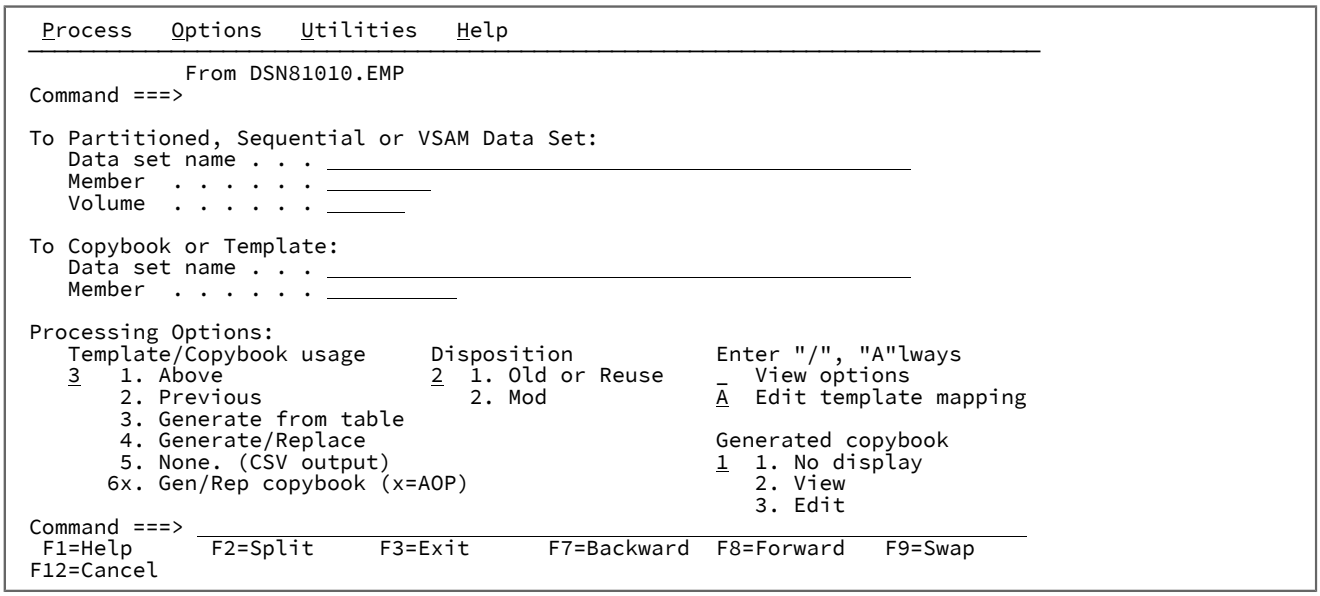
Related tasks

- Copying data to a VSAM or QSAM file on page 279
- Specifying a data set and a member name on page 52
- Selecting options on ZDT/Db2 panels on page 45
- Working with templates on page 74
- SAF-rule controlled auditing on page 71

Export Utility ("To") panel

You use the Export “To” panel to specify the name of the data set that contains the data to be exported, the (optional) name of a template data set, and processing options.

Panel and field definitions



Data set name (To Partitioned, Sequential or VSAM Data Set)

The name of the output data set.

Member (To Partitioned, Sequential or VSAM Data Set)

The output member name (if applicable).

Volume

The volume serial number if the data set is not catalogued.

Data set name (To Copybook or Template)

The name of the sequential or partitioned data set that contains the copybook or template that describes the data set. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member (To Copybook or Template)

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The Export Utility *"To"* panel provides five options for template processing:

1. **Above**

   Requires that you enter the name of a copybook or template in the **To Copybook or Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a copybook or template, processing continues using this template in place of the automatically-generated template for the table (see Template usage option 3).

2. Previous

Uses the last (previously used) template for this table.

3. Generate from input

Generates a template based on the input template, but which includes only those columns selected in the input template. If you have specified a data set name in the **To Copybook or Template** section of the panel, it is ignored. The generated template is not saved.

4. Generate and save

Generates a template based on the input template, but which includes only those columns selected in the input template. This option requires you to enter the name of a template data set (and optionally a member name) in the **To Copybook or Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. The generated template is saved.

5. None. (CSV output)

ZDT/Db2 does not use an output template. This option is only valid when the export data format is set to CSV.

Gen/Rep copybook (x=AOP)

Generates and saves a copybook that describes the exported data. This option requires you to enter the name of a copybook data set (and optionally a member name) in the *"To Copybook or Template"* section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. The generated copybook is saved, replacing any member with the same name.

- Specify 6A to generate and save a High Level Assembler copybook.
- Specify 60 to generate and save a COBOL copybook.
- Specify 6P to generate and save a PL/I copybook.

Disposition

The disposition for the data set that contains the exported data.

1. Old or Reuse

Specify '1' when the export data set exists, and the exported data is to replace any existing data.

2. Mod

Specify '2' in one of the following conditions:

- The export data does not exist, and needs to be created.
- The export data set exists, and the exported data is to be added to any existing data.

You can select the following options by entering a / or an “A”:

Edit options

Select this option to display the second of the Export Options panels. The default is not selected.

Edit template mapping

Use to edit the template for the partitioned, sequential, or VSAM data set to which you are exporting the data, before exporting the data. The default is not selected.

Note: Template mapping does not apply for CSV output. See the template usage option **5. None. (CSV output)**.

Parent panels

- Export Utility ("From") panel on page 698
- Column Selection/Edit panel on page 497

Child panels

To display this panel…	Use/do this
Template Mapping panel on page 900	Select **Edit template mapping**

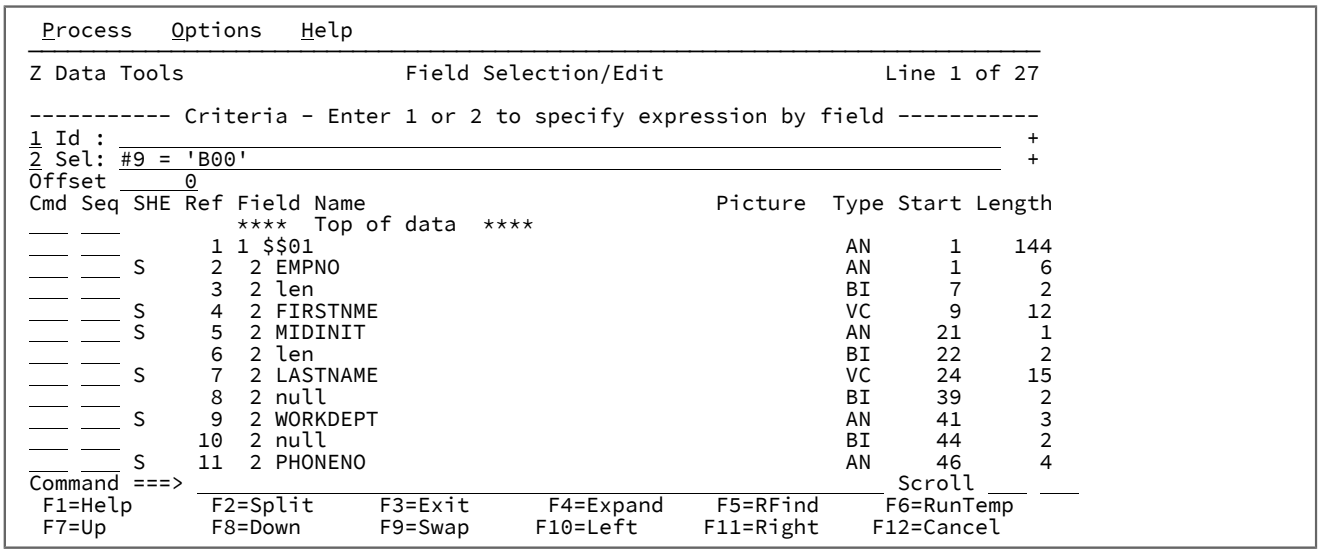
Related tasks

- Copying data to a VSAM or QSAM file on page 279
- Specifying a data set and a member name on page 52
- Exported data formats on page 285
- Selecting options on ZDT/Db2 panels on page 45
- Working with templates on page 74

Field Selection/Edit panel

You use the Field Selection/Edit panel to specify the records you want to select when you use the Import utility function (option 3.6).

Panel and field definitions



1 Id, 2 Sel

Use option 1 and 2 to specify record identification and selection criteria by field.

If an expression has not previously been updated by field, you can enter the expression on the adjacent line which is left and right scrollable with the Left function key (F10) and the Right function key (F11), and expandable with the Expand function key (F4). The *"+"* and *"-"* indicators at the end of the field denote the directions (right and left respectively) in which you can scroll.

If an expression has previously been entered by field, this line is protected. To change back to a freeform REXX expression, select the *"by field"* option (by entering `1` or `2` on the command line)) and delete the expression. When you return to this panel the line is available for input (unless it is a dynamic template in which case you must specify the expression).

Note: For more information about specifying record selection criteria, see the Z Data Tools User's Guide and Reference.

Cmd

Prefix command area. Used to enter a template editor prefix command.

Seq

Sequence field. Used to order fields for display in a Z Data Tools/Db2 editor session and printing data using the Print utility.

SHE

Select/Hold/Edit field. Shows the field status:

S

The field has been selected for processing.

H

The field has been selected to be held on the edit display.

E

The field has been previously edited to add information such as data create, field headings, and scrambling options.

Ref

Field reference. Shows the field reference number assigned to the field name. Field reference numbers are assigned to all fields in a record, and are used to identify fields in selection criteria expressions.

Field Name

Shows the level number and field name of the field. If the field is an array, the dimensions are shown in brackets after the field name. If the field has been selected for processing, the field name is highlighted.

Picture

For COBOL, shows the picture clause. For HLASM shows the DC or DS operand. For PL/I, shows the length and scale (if non zero) for binary and packed fields. Also shows the bit length for bit fields.

Type

Shows the data type of the field:

AN

Alphanumeric (includes alphabetic, and group items)

BI

Binary

BT

Bit

DB

DBCS

FE

Floating point (external)

FP

Floating point (internal)

G

Graphic

PD

Packed decimal (internal decimal)

VC

Variable Character

VD

Variable DBCS

VG

Variable Graphic

ZC

Character Null terminated

ZD

Zoned decimal (external decimal)

ZE

Zoned decimal edited

ZG

Graphic Null terminated

Z2

DBCS Null terminated

Start

Shows the start column of the field. For variable located fields the start location is based on the maximum length of the record.

Length

Shows the length of the field. For a level-01 field, if the record is variable-length, the length shown is the maximum record length.

Record identification criteria

Used to specify the criteria that ZDT/Db2 uses to identify this record type. ZDT/Db2 first uses the record length to identify record type.

For fixed-length records (as defined by the record definition in the template, not the data set attributes), the length of the record read must exactly match the length of the corresponding record definition in the template.

For variable-length records, the length of the record read must be equal to or greater than the minimum length and equal to or less than maximum length of the corresponding record definition in the template.

If ZDT/Db2 cannot match the length of a record to one of the record definitions in the template, the record is not selected for processing. If the length of a record matches more than one record definition in the template, and no further record identification criteria is specified, ZDT/Db2 uses the first matching record type in the template. Therefore, if the record length of each different record type in the file is not unique, you should use this field to specify some unique identification criteria.

Record selection criteria

Used to specify the criteria that ZDT/Db2 uses to select records for processing. You can use this field to limit the records of a given record type that you want to process. If you do not provide any record selection criteria, all records of this record type are selected for processing (if the record type itself has been selected for processing).

Parent panels

- Import Utility ("From") panel on page 719

Child panels

None.

Related tasks

- Copying data from a VSAM or QSAM file on page 251

ZDT/Db2 Release News panel

ZDT/Db2 displays the ZDT/Db2 Release News panel if you:

1. Select **Help** from the Action bar menu, and
2. Select **8. News about ZDT/Db2...** from the drop-down menu.

Foreign Keys for Table panel

ZDT/Db2 displays the Foreign Keys for Table panel if you use the FK line command against an item listed on the Tables, Views and Aliases panel on page 891.

Free Application Plan panel

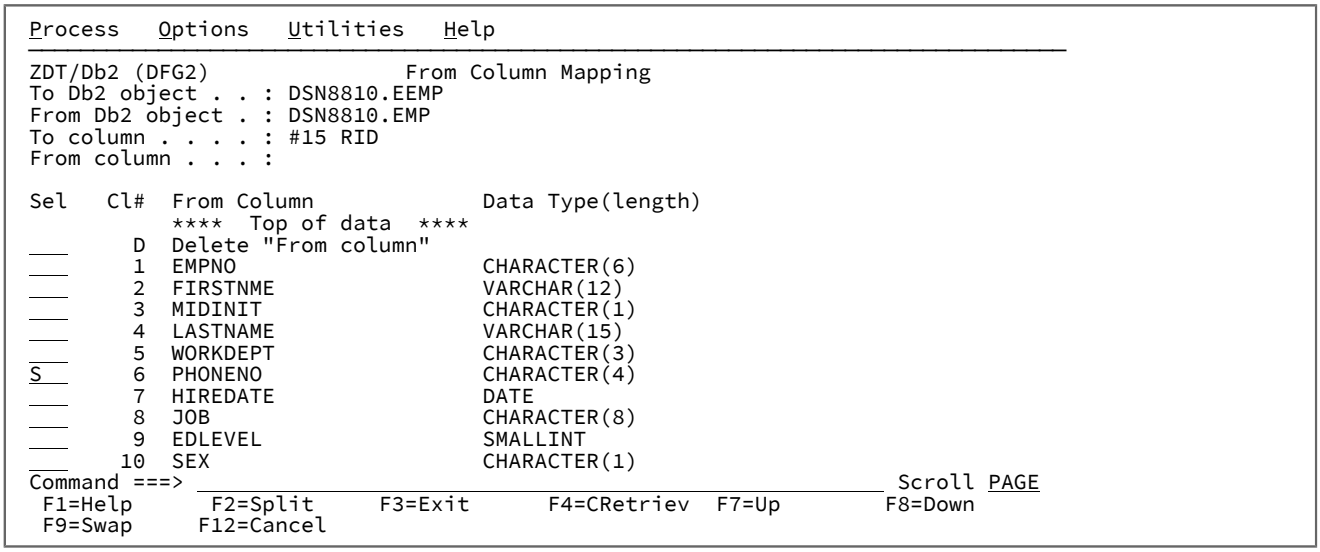
ZDT/Db2 displays the Free Application Plan panel if you use the FRE line command against an item listed on the Application Plans panel on page 473.

Free Package panel

ZDT/Db2 displays the Free Package panel if you use the FRE line command against an item listed on the Application Packages panel on page 469.

From Column Mapping panel

Panel and field definitions



Parent panels

- Template Mapping panel on page 900

Child panels

To display this panel…	Use/do this
Template Mapping panel on page 900	Press F3 or F12

Functions panel

You use the Functions panel to display information about the functions you have selected.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSROUTINES.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                      Functions
Location:                                                      Format TABL

SEL    SCHEMA   NAME     ORIGINEX          OWNER    ROUTINEX
       *        *        *                 *        *
----   #1--+--- #3--+--- #90-+----1----+- #2--+--- #89-+----1----+----2--
****  Top of data  ****
_____  SHRIKE2  DT1      System generated  SHRIKE2  Function/Cast Function
_____  SHRIKE2  CHAR     System generated  SHRIKE2  Function/Cast Function
_____  SHRIKE2  DT1      System generated  SHRIKE2  Function/Cast Function
_____  SHRIKES  CHAR80   System generated  SHRIKE2  Function/Cast Function
_____  SHRIKES  CHAR     System generated  SHRIKE2  Function/Cast Function
_____  SHRIKES  CHAR80   System generated  SHRIKE2  Function/Cast Function
_____  SHRIKES  DOCMATCH External         SHRIKES  Function/Cast Function
****  End of data  ****


Command ===> _____________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Schemas panel on page 812

Child panels

To display this panel…	Use/do this
	Line command A

To display this panel…	Use/do this
Alter External Scalar Function panel on page 454	
	Line command COM
See Comment panels on page 506	
	Line command CR
Create Function panel on page 536	
	Line command DR
See Drop panels on page 640	
	Line command G
Grant privileges panels on page 715	
	Line command GEN
Generate SQL From Db2 Catalog panel on page 712	
	Line command I
Details panels on page 636	
	Line command P
Privileges panels on page 768	
	Line command PARM
Routine Parameters panel on page 799	
	Line command PKG
Application Packages panel on page 469	
	Line command PS
	Line command R
Revoke privileges panels on page 798	
	Line command RDT

To display this panel…	Use/do this
Distinct Types panel on page 637	
	Line command ROW
Display Row panel on page 637	
	Line command SCH
Schemas panel on page 812	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

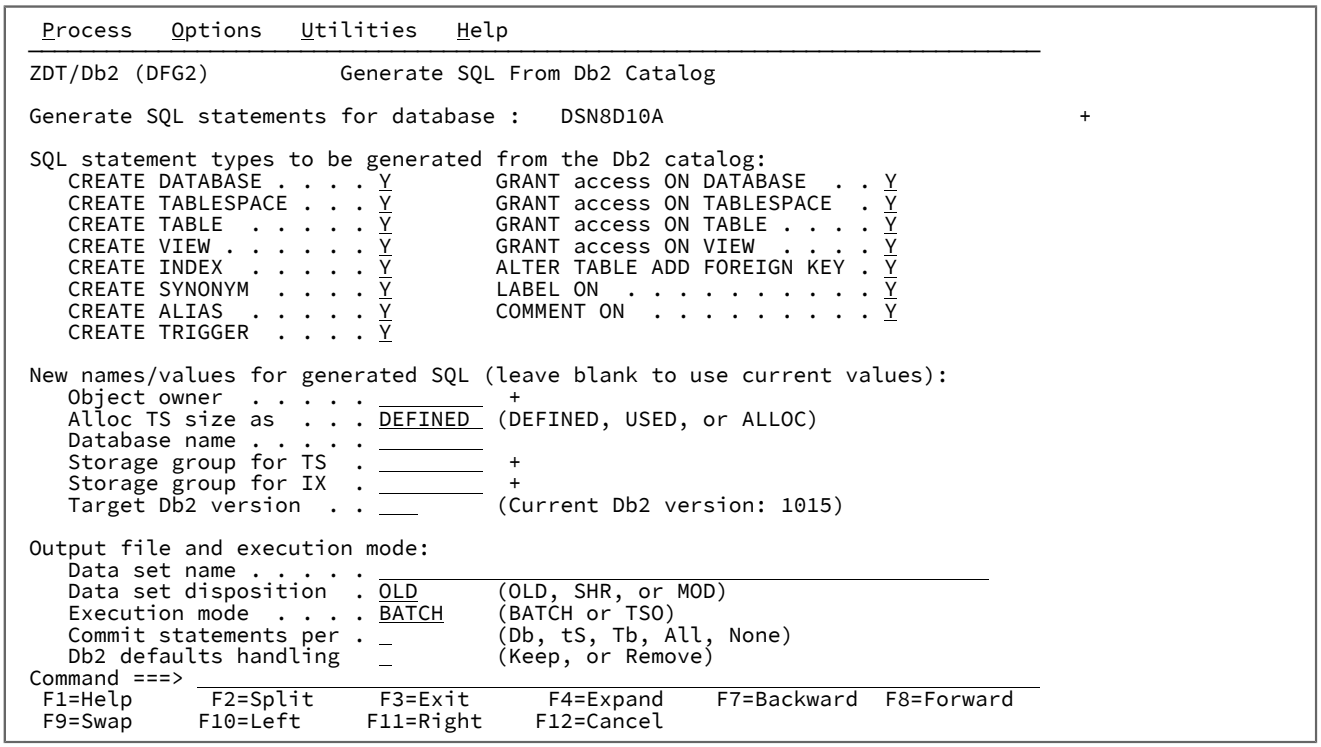
Related references

- SORT primary command on page 1003

Generate SQL From Db2® Catalog panel

You use the Generate SQL From Db2® Catalog panel to specify how the generation of SQL from the Db2® catalog should work. You can specify which SQL statement types ZDT/Db2 should generate, and optionally which names and values should be changed.

Panel and field definitions



Generate SQL statements for database

Specify `Y` for each SQL statement type you want ZDT/Db2 to generate, and `N` for each SQL statement type you do not want generated.

Object owner

New owner (authorization ID) of the objects.

Alloc TS size as

Specifies how to generate the primary quantity:

DEFINED

Use the size defined in the catalog

USED

Use the size that actually is used

ALLOCATED

Use the allocated size

USED and **ALLOCATED** require that the STOSPACE utility has been run for the storage groups for the objects being generated.

Database name

New database name for the objects.

Storage group for TS

New storage group for the table spaces.

Storage group for IX

New storage group for the indexes.

Target Db2® version

Db2® level of generated SQL statements.

Data set name

An existing valid SPUFI input data set name, or as `SYSOUT=x`. `SYSOUT=*` is the default if the field is blank.

Data set disposition

The disposition of the output data set.

Execution mode

Specify:

BATCH

Generates a batch job

TSO

Runs the generation online

Commit statements per

When to generate COMMIT statements:

D

Ffor each database

S

For each table space

T

Ffor each table

A

For all objects

N

Never

Db2® defaults handling

Specifies how Db2® default parameters should be handled:

K

Keep Db2® default parameters

R

Remove Db2® default parameters

Parent panels

- Databases panel on page 609
- Table Spaces panel on page 879
- Tables, Views and Aliases panel on page 891
- Indexes panel on page 723
- Schemas panel on page 812
- Distinct Types panel on page 637
- Functions panel on page 709
- Stored Procedures panel on page 852

Child panels

None.

Related tasks

- Reverse engineering on page 324

Grant privileges panels

These panels are:

- Grant Application Plan Privileges panel
- Grant Buffer Pool Privileges panel
- Grant Collection Privileges panel
- Grant Column Privileges panel
- Grant Database Privileges panel
- Grant Distinct Type Privileges panel
- Grant Function Privileges panel
- Grant Package Privileges panel
- Grant Schema Privileges panel
- Grant Sequence Privileges panel
- Grant Storage Group Privileges panel
- Grant Stored Procedure Privileges panel
- Grant System Privileges panel

- Grant Table Privileges panel
- Grant Table Space Use Privileges panel

ZDT/Db2 displays one of these panels if you use the Privileges utility function (3.5) to grant privileges for an object type, or if you specify the G line command against an object displayed in a list of Db2® objects.

Format of grant privilege panels

Each grant privilege panel shows:

- The available privileges for the object type, with a corresponding single-character entry field in which you can specify one of the following values:

  Y

  Grants the specified privilege to the grantee, but does not permit the grantee to grant this privilege to other users.

  G

  Grants the specified privilege to the grantee, and permits the grantee to grant this privilege to other users.

  A blank in the field indicates that the grantee does not have the specified privilege.

  Note: To change an *existing* privilege, you must first revoke the existing privilege before granting the new privilege (if applicable) as described in this section.

- A field (for some types of privilege there are two fields), that you use to specify the name of the object for which the privilege is to be granted.
- A **To** field. Use this to specify the SQLID of the user to whom the privilege is to be granted.

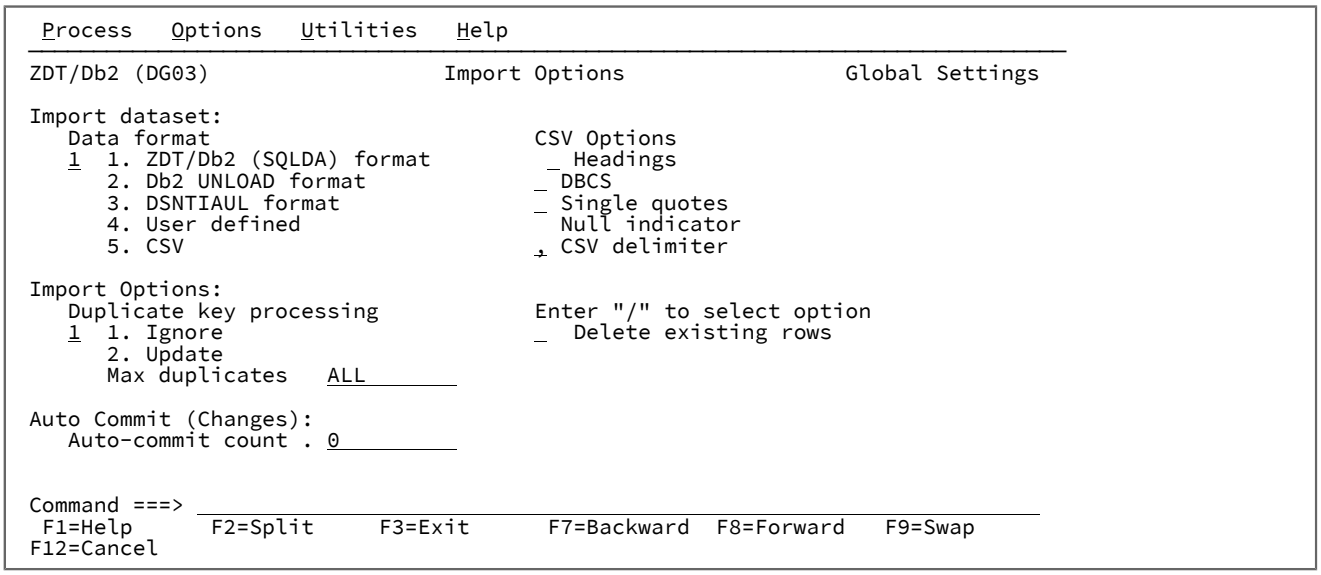
Related tasks

- Displaying privileges on page 342
- Working with lists of Db2 objects on page 303
- Granting privileges on page 342
- Revoking privileges on page 344

Import Options panel

You use the Import Options panel to set either global or local import options, depending on how the panel is displayed.

Panel and field definitions



Data format

Allows you to specify the data format that corresponds to the template or copybook used:

1. ZDT/Db2 (SQLDA) format
2. Db2® UNLOAD format
3. DSNTIAUL format
4. User defined
5. CSV

The default is 1. (ZDT/Db2 (SQLDA) format).

Duplicate key processing

1. Ignore

Ignores duplicate rows. This is the default.

2. Update

Updates duplicate rows when Db2® returns a duplicate row error (SQLCODE -803) in response to an INSERT SQL statement. When you select this option, ZDT/Db2 attempts to update the existing row. All columns of the table that are part of at least one unique index are used to identify the row. No comparison is made between the imported row and the existing row prior to the update operation (that is, the update is done regardless of whether the imported row and existing row are the same). When you use this option it is recommended that you specify a Db2® table name, rather than a view name. ZDT/Db2 does not attempt to update existing rows in a view defined on a view.

Max duplicates

The maximum number of duplicates permitted.

Valid values: A number in the range 0–9999999999, or ALL. The default is ALL.

Auto-commit count

Use this option to take progressive Db2® commit points as records are imported. Any positive value entered will cause ZDT/Db2 to issue a Db2® commit automatically after the specified number of successful Db2® changes. A successful Db2® change is either the successful insert of a record; or the successful update of a row when a duplicate key error is encountered on the insert. The change counter is reset after each Db2® commit is issued, so multiple Db2® commit points may be issued. Specify a zero value if no progressive Db2® commit points are required; in this case the Db2® commit is deferred until the end of the import utility and any error encountered during the import will cause all changes to be backed out. The default for this option is 0.

CSV Options

Headings

Select this option if the CSV data set to be imported includes headings.

DBCS

Select this option if the CSV data set to be imported includes double-byte data (DBCS) characters wrapped in shift-in (x'0E') and shift-out (x'0F') characters.

Single quotes

Select this option if the CSV data set to be imported has single quote encapsulation.

Null indicator

Use this field to specify the null indicator value that was used to represent a null value in the CSV input data set. A nullable column will be set to null when the corresponding CSV value has:

- A non-blank null indicator value.
- A null CSV value. For example, two successive delimiters ',,'.

CSV delimiter

Use this field to specify an alternate delimiter value to the comma "," character.

Delete existing rows

Deletes all the rows in the target table before importing. The default is not selected; that is, no rows are deleted from the target table.

Parent panels

- Set Utility Options panel on page 840
- Import Utility ("From") panel on page 719
- Import Utility ("To") panel on page 720

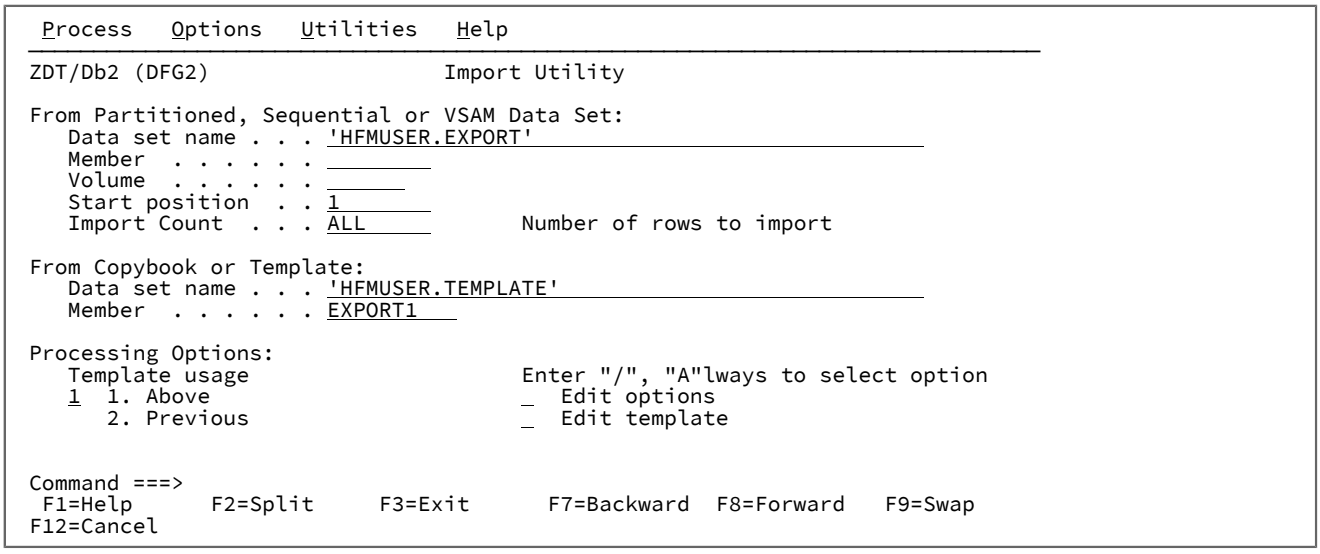
Child panels

None.

Related tasks

Import Utility ("From") panel

Panel and field definitions



You can select the following options by entering a / or an “A” (see Selecting options on ZDT/Db2 panels on page 45):

Edit template

Use to edit the template for the table from which you are importing the data, before importing the data. The default is not selected. For more details, see Working with templates on page 74.

Edit options

Select this option to display the Import Options panel. The default is not selected.

Parent panels

- Utility Functions panel on page 915

Child panels

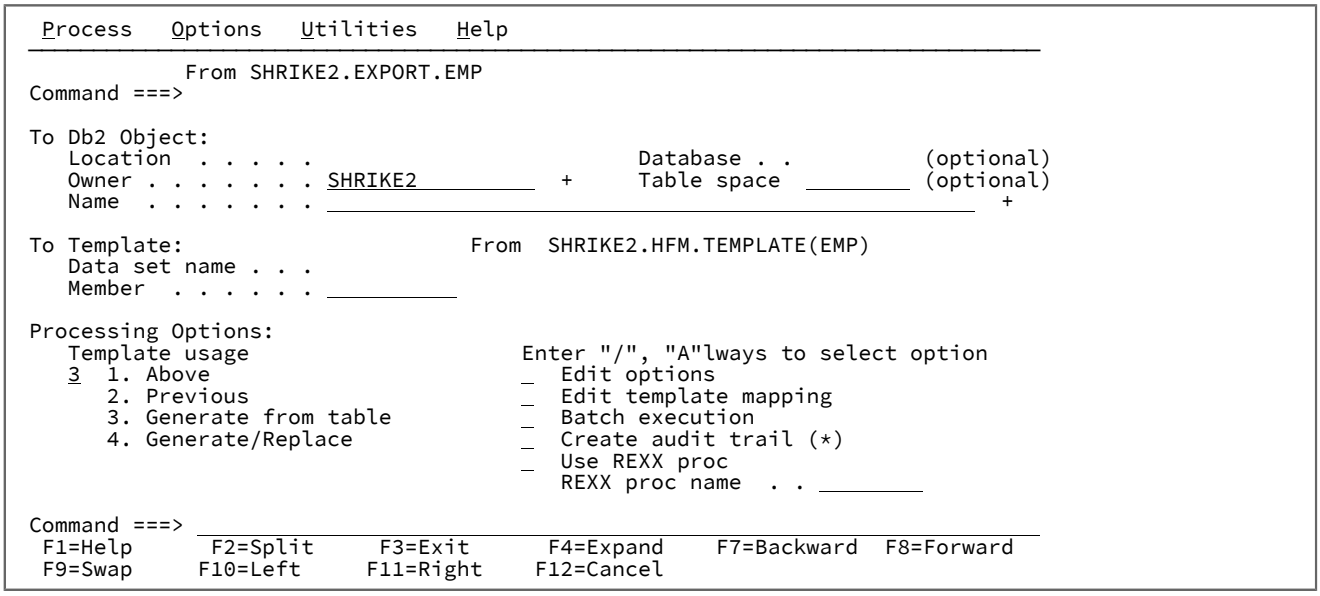
To display this panel…	Use/do this
Import Options panel on page 716	Select **Edit options**
Field Selection/Edit panel on page 704	Select **Edit template**

Related tasks

- Selecting options on ZDT/Db2 panels on page 45
- Working with templates on page 74

Import Utility ("To") panel

Panel and field definitions



Location

The location name for the Db2® object. You can specify a generic name to select from a list. If you enter a value in this field, synonym object names are not recognized or included in the selection list.

Database

The database name for the Db2® object. You can use this field to optionally specify a database to further define the object to be processed. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified database.
- Synonym names are not recognized or included in the selection list.

Owner

The name of the owner of the object. You can specify a generic name to select from a list. To specify a synonym in the object **Name** field, either leave this field blank, or specify a generic name that includes the current SQLID or specify the current SQLID. Synonyms are not recognized or included on the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Table space

The table space name for the Db2® object. To further define the object to be processed, you can use this field to optionally specify a table space. If you enter a value in this field:

- The values displayed in the **Owner** and **Name** selection lists are limited to those values defined within the specified table space.
- Synonym names are not recognized, nor included in the selection list.

Name

The name of the object name to be processed. You can specify a generic name to select from a list. To specify a synonym in this field, type the name of the synonym and ensure that the **Owner** field is blank, contains a generic name that includes the current SQLID, or contains the current SQLID. Synonyms are not recognized or included in the table selection list if any of the **Table space**, **Database**, or **Location** fields are non-blank.

If the **Translate Db2® object names** option is selected, lowercase characters are converted to uppercase. If this option is not selected, lowercase characters are not converted to uppercase; in this situation, a lowercase owner name such as *"sysibm"* does not match *"SYSIBM"* in the Db2® catalog.

Data set name

The name of the data set or PDS member containing the template for the *"To"* table. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

The Import Utility *"To"* panel provides four options for template processing:

1. Above

Requires that you enter the name of a template in the **To Template** section of the panel.

2. Previous

Uses the last (previously used) template for this table.

3. Generate from table

Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a data set name in the **To Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. Generate/Replace

Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the data set specified in the **To Template** section of the panel. If the template already exists, it is replaced and any prior customization is lost.

You can select the following options by entering a / or an "A":

Edit options

Select this option to display the Import Options panel. The default is not selected.

Edit template mapping

Use to edit the template for the table to which you are importing the data, before importing the data. The default is not selected.

Batch execution

Builds the necessary JCL to perform the Import utility in a batch job. The default is not selected.

Create audit trail

Note: "(*)" is appended to the option name for SAF-controlled auditing when required by SAF rules.

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Use REXX proc

Use this option to perform either of these actions:

- Enter a temporary REXX procedure for one-time use by entering an asterisk (*). ZDT/Db2 displays a REXX edit panel that you can use to create a new REXX procedure. The default is not selected.
- Specify the name of the member containing the REXX procedure you want to use. The member must belong to the PDS allocated to ddname HFMEXEC. You can enter any of the following:
  - The name of the member.
  - A member name pattern (other than *) to list all matching members. You can then select the required member by entering an `s` in the **Sel** field. A member name pattern can consist of any characters that are valid in a member name and the following two special pattern characters:

    asterisk (*)

    Represents any number of characters. As many asterisks as required can appear anywhere in a member name. For example, if you enter `*d*`, a list of all members in the data set whose name contains "d" is displayed.

percent sign (%)

A place-holding character representing a single character. As many percent symbols as necessary can appear anywhere in a member name. For example, if you enter `%%%`, a list of all members in the data set whose name is four characters in length is displayed.

Note: The default for the member name is blank. If you select this option but leave the **Use REXX proc** field blank, ZDT/Db2 displays a member name list. You can then select the required member by entering `s` in the **Sel** field.

Parent panels

- Import Utility ("From") panel on page 719
- Field Selection/Edit panel on page 704

Child panels

To display this panel…	Use/do this
Import Options panel on page 716	Select **Edit options**

Related tasks

- Working with templates on page 74
- Selecting options on ZDT/Db2 panels on page 45
- Using a REXX procedure with the Import utility on page 257
- Enhancing the Import utility using a REXX procedure on page 268
- SAF-rule controlled auditing on page 71

Index Parts panel

ZDT/Db2 displays the Index Parts panel if you use the PT line command against an item listed on the Indexes panel on page 723.

Indexes panel

You use the Indexes panel to list index object types in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSINDEXES.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                      Indexes
Location:                                                    Format TABL

                                                                          +
SEL    CREATOR       NAME                TBNAME              TBCREATOR    UNIQUEE
       *             *                   *                   *            *
----   #2--+----1--  #1--+----1----+---  #3--+----1----+---  #4--+----1--  #48-+--
****  Top of data  ****
_____  CARLAND       XPMR91932           PMR91932            CARLAND      Unique
_____  DSN8810       XACT1               ACT                 DSN8810      Unique
_____  DSN8810       XACT2               ACT                 DSN8810      Yes
_____  DSN8810       XCONA1              TCONA               DSN8810      Yes
_____  DSN8810       XDEPT1              DEPT                DSN8810      Unique
_____  DSN8810       XDEPT2              DEPT                DSN8810      Duplica
_____  DSN8810       XDEPT3              DEPT                DSN8810      Duplica
_____  DSN8810       XDSPTXT1            TDSPTXT             DSN8810      Yes
_____  DSN8810       XEMP1               EMP                 DSN8810      Unique
_____  DSN8810       XEMP2               EMP                 DSN8810      Duplica
_____  DSN8810       XEMPPROJACT1        EMPPROJACT          DSN8810      Yes
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Databases panel on page 609
- Tables, Views and Aliases panel on page 891
- Columns panel on page 503
- Application Packages panel on page 469

Child panels

To display this panel...	Use/do this
	Line command A
Alter Index panel on page 455	
	Line command COL
Columns in Index panel on page 505	
	Line command CR
Create Index panel on page 542	
	Line command D
Databases panel on page 609	
	Line command DR
See Drop panels on page 640	
	Line command GEN
Generate SQL From Db2 Catalog panel on page 712	
	Line command I
Details panels on page 636	
	Line command ICS
Recovery Information panel on page 787	
	Line command PKG
Application Packages panel on page 469	
	Line command PL
Application Plans panel on page 473	
	Line command PT

To display this panel...	Use/do this
Index Parts panel on page 723	
	Line command ROW
Display Row panel on page 637	
	Line command T
Tables, Views and Aliases panel on page 891	
	Line command UTL
Db2 Utilities panel on page 627	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

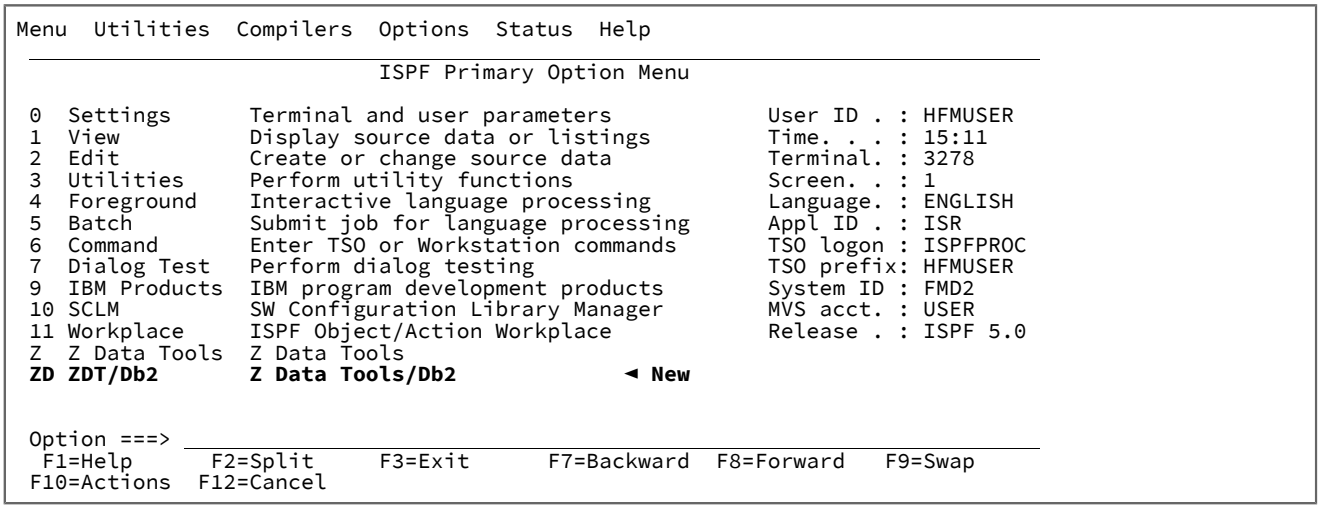
- SORT primary command on page 1003

Indexes and Columns for panel

ZDT/Db2 displays the Indexes and Columns for panel if you use the XC line command against an item listed on the Tables, Views and Aliases panel on page 891.

ISPF Primary Option panel

Panel and field definitions



Parent panels

- None.

Child panels

To display this panel…	Use/do this
Primary Option Menu panel on page 760	Select option for ZDT/Db2

Related tasks

ISPF Settings panel

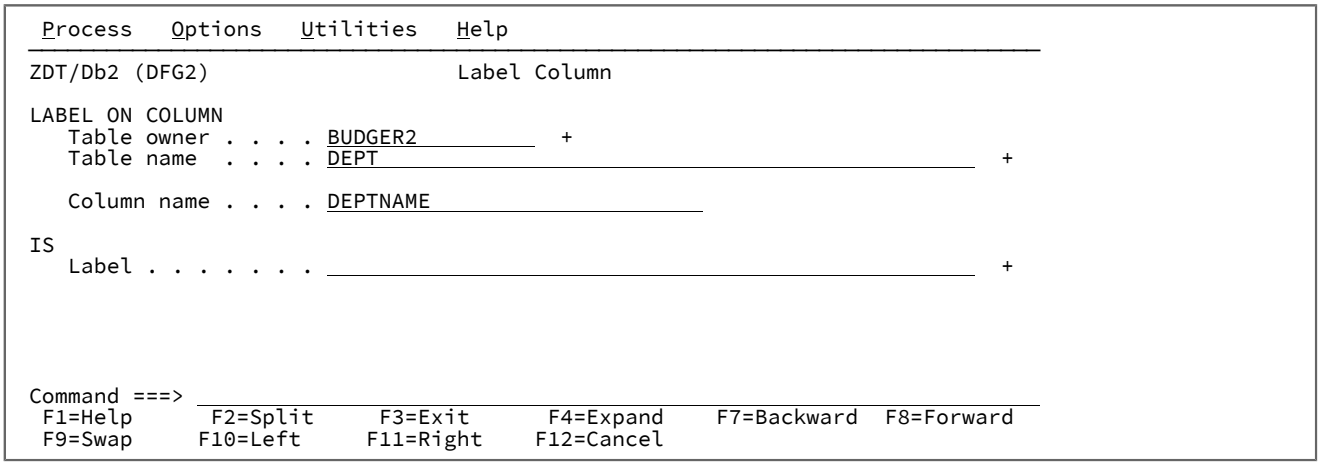
You use the ISPF Settings panel display and modify selected ISPF parameters.

Related tasks

- ISPF settings (option 0.11) on page 67

Label Column panel

Panel and field definitions



Parent panels

- Columns panel on page 503
- Constraint Columns for panel on page 509
- Columns in Table panel on page 505

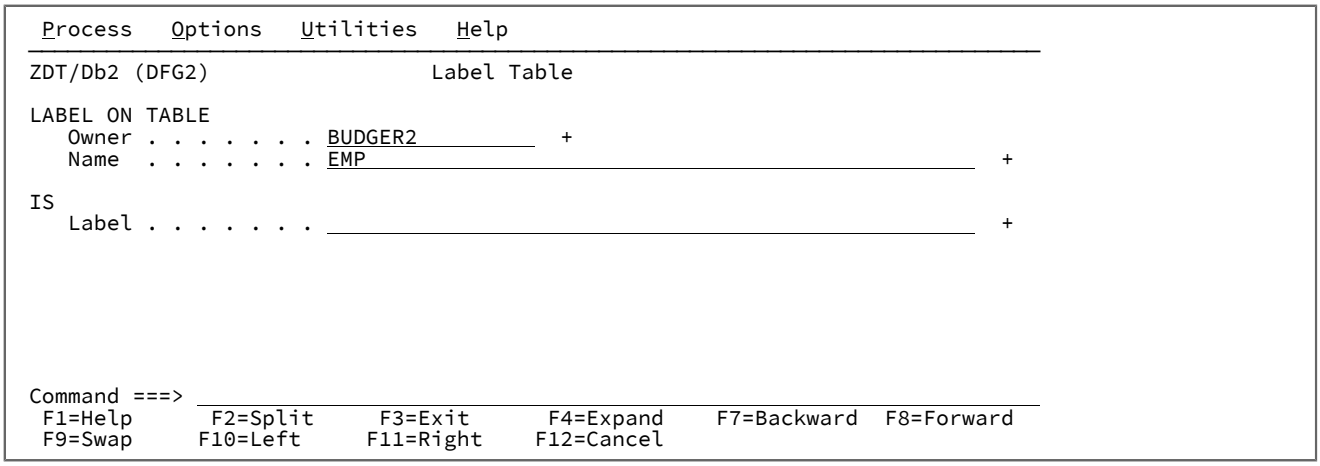
Child panels

None.

Related tasks

Label Table panel

Panel and field definitions



Parent panels

- Tables, Views and Aliases panel on page 891

Child panels

None.

Related tasks

LISTCAT Output Display panel

ZDT/Db2 displays the LISTCAT Output Display panel when you use the LC line command on the Recovery Information panel or the Table Space Parts panel.

LISTDEF Options panel

You use the LISTDEF Options panel to collect information used in generating LISTDEF statements and the proper JCL statement to access the LISTDEF library in a utility batch job.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                LISTDEF Options                Global Settings

Enter RESET to restore installation defaults.
LISTDEF Library Option:
   JCL DD statement:              No DD stmt is generated if field is blank
      //SYSLISTD DD _____________________________________________________________
      //          DD _____________________________________________________________
      //          DD _____________________________________________________________
      //          DD _____________________________________________________________
Initial Utility Statement Option:
   Utility statement:             No statement is generated if field is blank
      LISTDEF ____________________________________________________________________
      _____________________________________________________________________________
      _____________________________________________________________________________
      _____________________________________________________________________________
   Additional statements:
      Enter "/" to select options
      _  Use additional statements
      _  Edit additional statements
Command ===> ________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

JCL DD statement

The information used to generate the JCL DD statement. This information is:

DD name

A read-only field. This holds the value specified in the **LISTDEFDD DD name** field of the OPTIONS Options panel (OPTIONS Options panel on page 750). If the library field is blank then no DD statement is generated in the utility batch job. The default is blank.

Text values

The values entered on these four lines are used directly to generate the DD statement, provided that the **DD name** field is not blank.

Utility statement

This value is used to generate the initial utility statements. If the field is blank, and the **Use additional statements** option is not selected, then no LISTDEF statement is generated into the utility batch job. If the field is non-blank, then the specified LISTDEF is generated with any additional LISTDEF statements supplied when the Use additional statements option is selected. The default is blank.

Use additional statements

If this option is set then more than one LISTDEF statement may be generated into the utility batch job. Select the **Edit additional statements** option to edit additional statements. Only LISTDEF specifications that have non-blank input fields are used in the generation. The default is not selected.

If this option is not selected, then only a non-blank value in the **Utility statement** field is used to generate the LISTDEF statement.

Edit additional statements

When you select this option and press Enter, the Additional LISTDEF Statements panel (Additional LISTDEF Statements panel on page 449) is displayed. After you have entered values on this panel and exited the panel, this option is deselected. The additional statements are used when you select the Use additional statements option. The default is not selected.

The LISTDEF Library Option panel has an installation-defined default. You can restore the default for the library option by entering the RESET command or pressing the RESET PF key. Each subsystem has its own installation-defined defaults. RESET also clears the **Use additional statements** option and any values in the **Utility statement** fields.

For more information, see LISTDEF and OPTIONS statements in *DB2 Utility Guide and Reference*.

Child panels

To display this panel…	Use/do this
Additional LISTDEF Statements panel on page 449	Select the Edit additional statements option.

Related tasks

- RESET primary command on page 994
- Setting options for Db2 utilities on page 348
- Db2 Utility LISTDEF options (option 3.UL) on page 69

LOAD from panel

You use the LOAD Utility: “To” template panel to specify the template for the target Db2® table when you use the LOAD utility.

Panel and field definitions

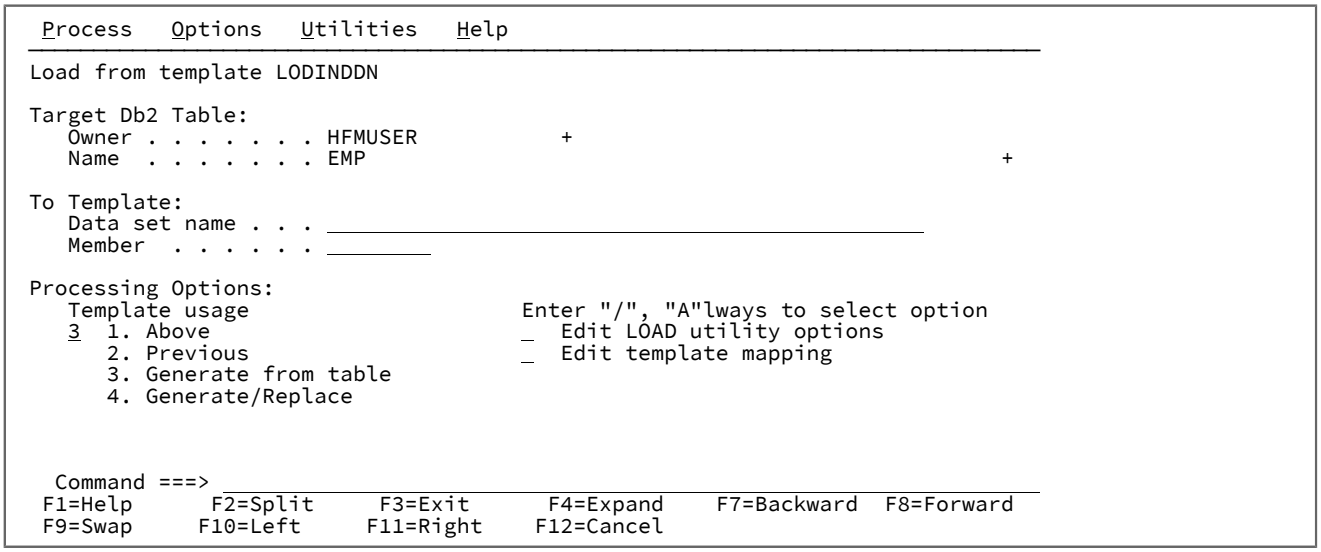


Table owner

The name of the owner for the target table. You can specify a generic object name to select from a list. This field is mandatory.

Table name

The name of the target table. You can specify at generic object name to select from a list. This field is mandatory.

Data set name

The name of the data set or PDS member containing the template for the *“To”* object. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The LOAD Utility: "To" template panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template in the **To Template** section of the panel.

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a data set name in the **To Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. **Generate/Replace**

   Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the data set specified in the **To Template** section of the panel. If the template already exists, it is replaced and any prior customization is lost.

You can select the following options by entering a / or an "A":

Edit load utility options

Displays the Load Utility Options panel

Edit template mapping

Use to edit the template for the table to which you are importing the data, before importing the data.

Parent panels

- LOAD Utility - Using Templates panel on page 735

Child panels

To display this panel…	Use/do this
LOAD Utility Options panel on page 737	Select **Edit LOAD utility options**
Template Mapping panel on page 900	Select **Edit template mapping**

Related tasks

- LOAD utility on page 357
- Working with templates on page 74
- Selecting options on ZDT/Db2 panels on page 45

LOAD Utility panel

You use the LOAD Utility panel to specify details about the data you want to load.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DB26)                   LOAD Utility

Input Data:
   Data set name . . _____________________________________________
   Member  . . . . . ___________
   Volume  . . . . . _______

Utility Control Statements:
   Data set name . . _____________________________________________________
   Member  . . . . . _________

Target Db2 Table:
   Table owner . _________                  Database . . _________ (optional)
   Table name  . ____________________        Table space  _________ (optional)

   Select Option:
   1  1. The data is described by utility control statements
      2. The data is described by a template or copybook

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Data set name (Input Data)

A sequential or partitioned data set that contains the data to be loaded into the Db2® table. Specify a generic data set name to select from a list. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member (Input Data)

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume

The volume serial number if the data set is not cataloged.

Data set name (Utility Control Statements)

A sequential or partitioned data set that contains the Db2® load utility control statements. Specify a generic data set name to select from a list. It can be a fully-qualified data set name or a pattern. The name may include

a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member (Utility Control Statements)

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Table owner

The name of the owner for the target table. You can specify a generic object name to select from a list. This field is mandatory.

Database

A fully or partially-qualified database name to limit the list of object names to the specified databases.

Table name

The name of the target table. You can specify at generic object name to select from a list. This field is mandatory.

Table space

A fully or partially-qualified table space name to limit the list of object names to the specified table spaces.

1. The data is described by utility control statements

Use the supplied Db2® load utility control statements when generating a Db2® load utility batch job.

2. The data is described by a template or copybook

Use a template or copybook when generating a Db2® load utility batch job.

Parent panels

- Db2 Utilities panel on page 627

Child panels

To display this panel…	Use/do this
LOAD Utility - Using Templates panel on page 735	Select option 2

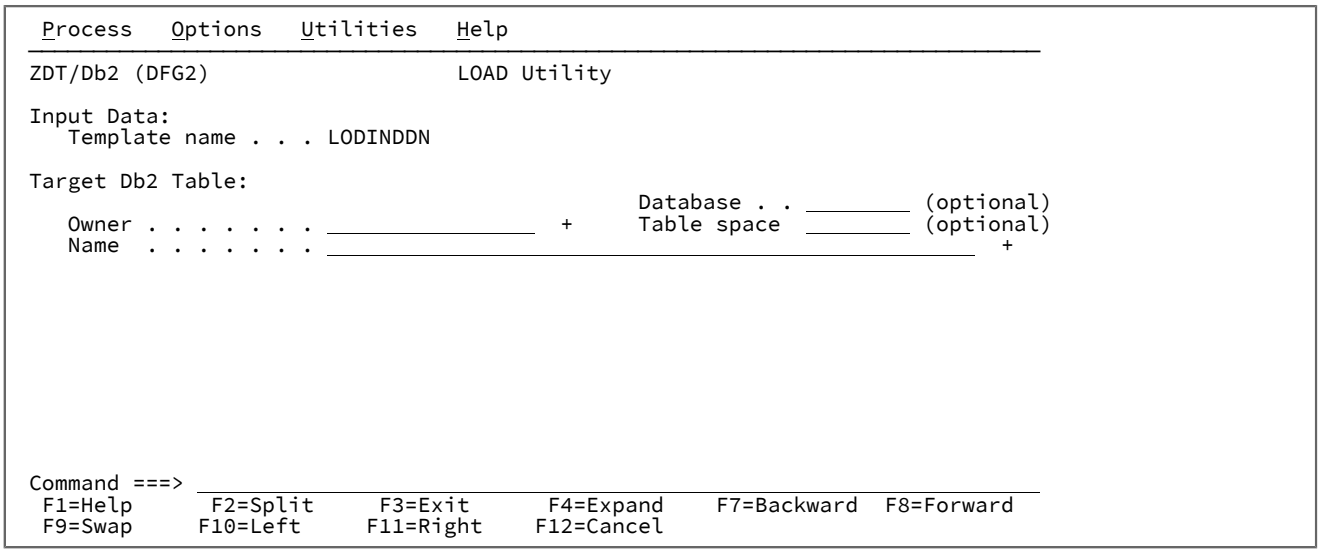
Related tasks

- LOAD utility on page 357

LOAD Utility panel using Db2® templates

You use the LOAD Utility using Db2® templates panel to load new data into tables and build or extend any indexes defined on those tables.

Panel and field definitions



Parent panels

- Db2 Utilities panel on page 627

Child panels

To display this panel...	Use/do this
LOAD Utility - Using Templates panel on page 735	Press Enter

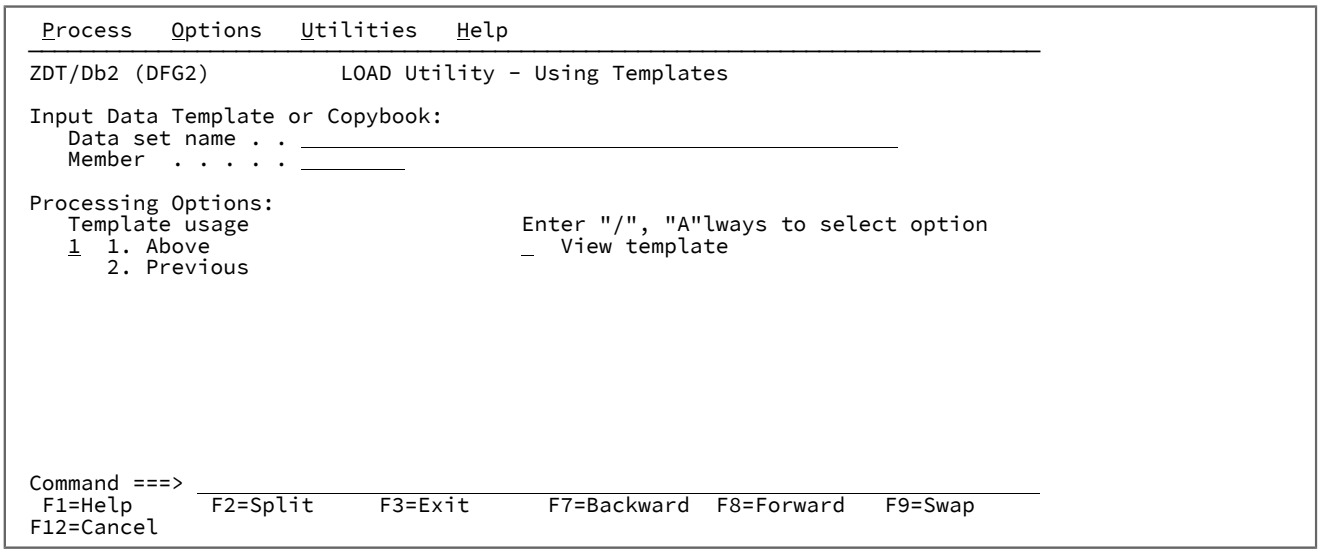
Related tasks

- LOAD utility on page 357

LOAD Utility - Using Templates panel

You use the LOAD Utility - Using Templates panel to specify the *"From"* copybook or template that describes the input data.

Panel and field definitions



Data set name

A sequential or partitioned data set that contains the data to be loaded into the Db2® table. Specify a generic data set name to select from a list. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Template usage

The LOAD Utility - Using Templates panel provides two options for template processing:

1. **Above**

   Requires that you enter the name of a template in the **Input Data Template or Copybook** section of the panel.

2. **Previous**

   Uses the last (previously used) template for this table.

You can select the following option by entering a / or an "A":

View template

Select this option to edit the *"To"* template.

Parent panels

- LOAD Utility panel on page 733

Child panels

To display this panel...	Use/do this
Record Type Selection panel on page 780	Select **Edit template**
LOAD from panel on page 731	Press Enter

Related tasks

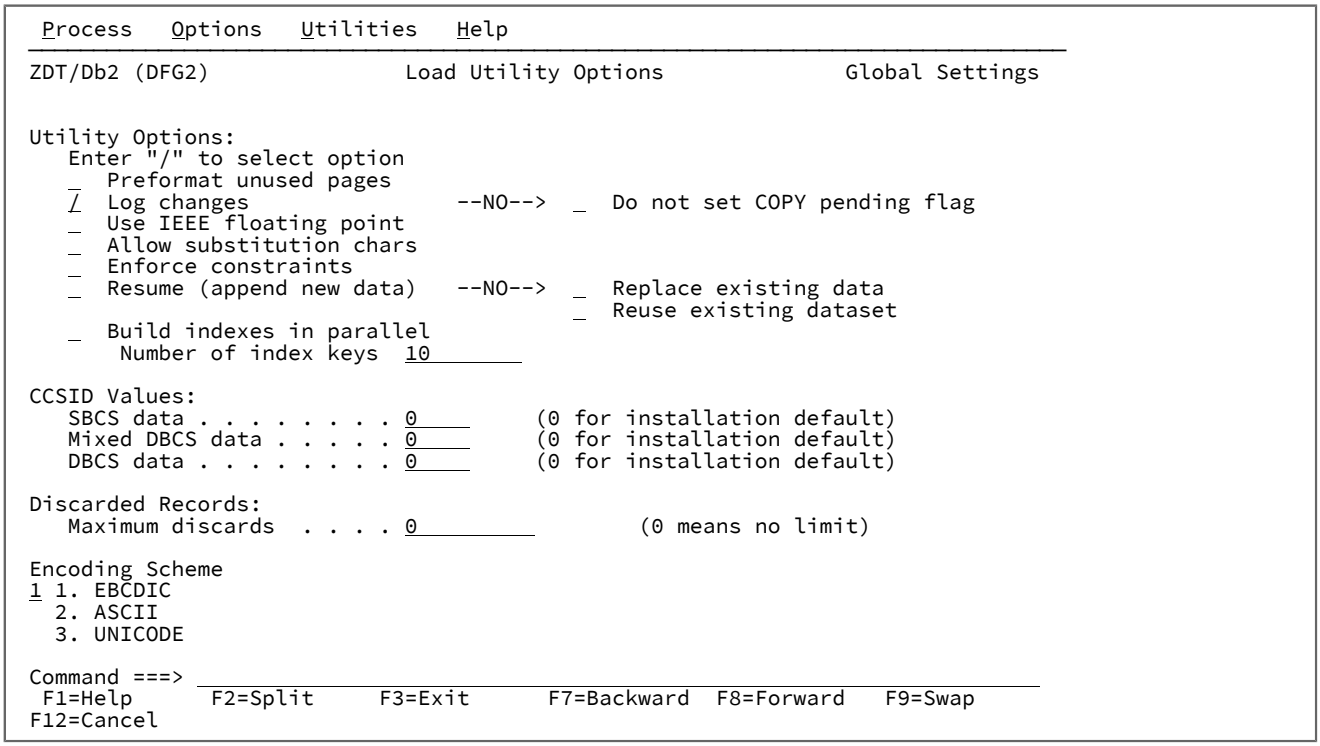
- LOAD utility on page 357
- Selecting options on ZDT/Db2 panels on page 45

LOAD Utility Options panel

You use the LOAD Utility Options panel to specify the options that apply when you use the LOAD utility to transfer data.

These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



Preformat unused pages

Indicates whether the allocated file space for a table is preformatted to enable quicker INSERT processing. The default is not selected.

Log changes

Indicates whether Db2® logs details in the Db2® log of the rows that are loaded. The default is selected.

Do not set COPY pending flag

If **Log changes** is not selected, indicates whether Db2® is *not* to set the table space in COPY pending status. The default is not selected.

Use IEEE floating point

Indicates whether floating-point numbers are in IEEE Binary Floating Point (BFP) format. The default is not selected.

Allow substitution chars

Indicates whether the LOAD utility accepts substitution characters in a string. The default is not selected.

Enforce constraints

Indicates whether the LOAD utility enforces check constraints and referential constraints. The default is selected.

Resume (append new data)

Indicates whether records are loaded into an empty or non-empty table space, without existing data being deleted. The default is not selected.

Replace existing data

If **Resume (append new data)** is not selected, indicates whether the table space and all its indexes are reset to empty before records are loaded. The default is not selected.

Reuse existing dataset

If **Resume (append new data)** is selected, indicates whether the LOAD utility logically resets and reuses Db2-managed data sets without deleting and redefining them. The default is not selected.

Build indexes in parallel

Indicates whether index keys are sorted in parallel during LOAD. The default is not selected.

Number of index keys

An estimated number of index keys to sort. The default is 10.

SBCS data

The coded character set for SBCS data in the input file. The default is 0.

Mixed DBCS data

The coded character set for mixed DBCS data in the input file. The default is 0.

DBCS data

The coded character set for DBCS data in the input file. The default is 0.

Maximum discards

The maximum number of discarded records allowed before LOAD abends. The default is 0.

Note: ZDT/Db2 does not attempt to validate every possible combination of load utility options. In particular, some combinations of options fail given the current state of the Db2® object being loaded. This can mean that the Db2® load utility job will fail if an invalid combination of options has been specified. If this occurs examine the failing messages for the Db2® LOAD utility job and respond accordingly.

Encoding Scheme

Indicates the encoding scheme for the data to be loaded. The encoding scheme applies to character and graphic data only.

1. **EBCDIC**

   The data is encoded by using an EBCDIC CCSID. This is the default.

2. **ASCII**

   The data is encoded by using an ASCII CCSID.

3. UNICODE

The data is encoded in unicode (either CCSID 1208 or CCSID 1200).

Parent panels

- None.

Child panels

None.

Related tasks

- Db2 LOAD utility options (option 3.L) on page 69
- LOAD utility on page 357

Manage Db2® Privileges panel

You use the Manage Db2® Privileges panel to display, grant, and revoke Db2® privileges.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)              Manage Db2 Privileges

   blank Display privileges                 G Grant privileges
       R Revoke privileges

Object Identification Criteria:
   Location  . . . . . _________________        Enter * for list
   Owner . . . . . . . _________________  +
   Name  . . . . . . . ____________________________________________________ +
   In  . . . . . . . . _____________________  + Database/table/collection/schema

Processing Options:
   Object Type
   3   1. Database          7.  Collection         13. Buffer pool
       2. Table space       8.  Schema             14. System
       3. Table             9.  Type               15. Sequence
       4. Column            10. Function
       5. Plan              11. Stored procedure
       6. Package           12. Storage group
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Location

The location of the objects you want listed.

Owner

The name of the person who created the Db2® object.

Name

The name of the Db2® object.

In

An optional field you can use to qualify certain object types.

Object Type

The type of Db2® object against which privilege operations are to be performed. The default is 3 (**Tables**).

Parent panels

- Utility Functions panel on page 915

Child panels

To display this panel…	Use/do this
Privileges panels on page 768	Blank command line
Grant privileges panels on page 715	*"G"* on command line
Revoke privileges panels on page 798	*"R"* on command line

Related tasks

- Managing Db2 privileges on page 339

Object List Line Commands panel

ZDT/Db2 displays the Object List Line Commands panel if you use the ? line command against an item listed on one of these panels:

- Application Plans panel on page 473
- Collections panel on page 482
- Columns panel on page 503
- Database Request Modules panel on page 607
- Databases panel on page 609
- Distinct Types panel on page 637
- Functions panel on page 709
- Indexes panel on page 723
- Application Packages panel on page 469
- Schemas panel on page 812
- Stored Procedures panel on page 852
- Synonyms panel on page 855

- Table Spaces panel on page 879
- Tables, Views and Aliases panel on page 891
- Triggers panel on page 903

Related tasks

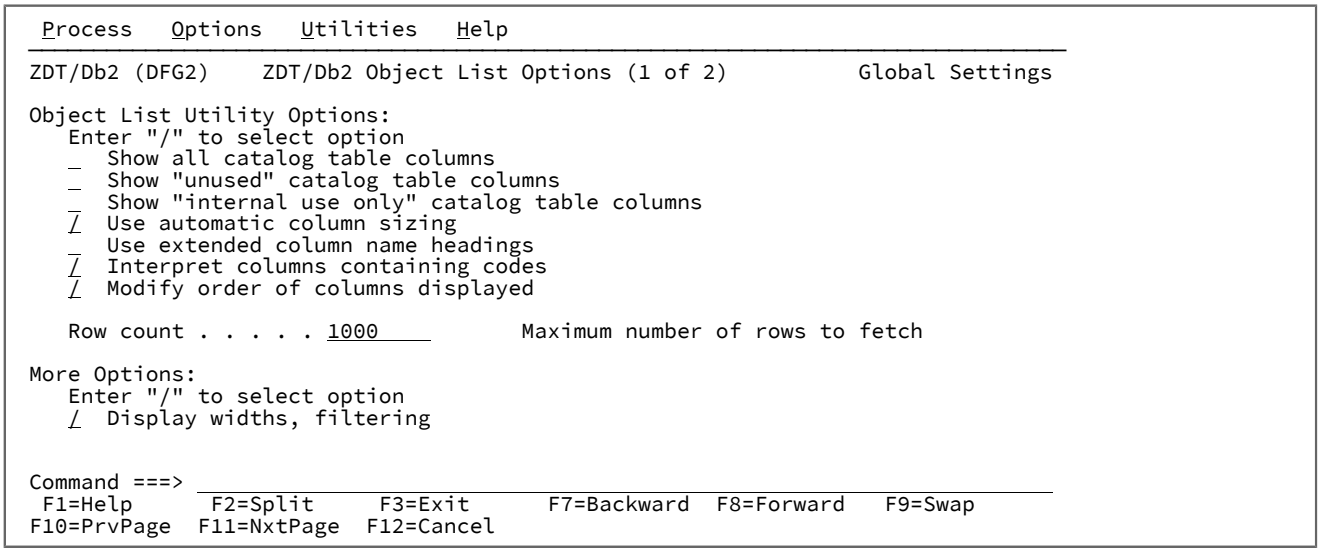
- Line command query (?) on page 320

Object List Options (1 of 2) panel

You use the ZDT/Db2 Object List Options (1 of 2) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 123. ZDT/Db2 Object List Options (1 of 2) panel



Show all catalog table columns

Display of catalog table information includes every available column, including any columns marked as "unused" or "internal use only" in the *DB2 for z/OS SQL Reference*. This option overrides any settings for the **Show "unused" catalog table columns** and **Show "internal use only" catalog table columns** options.

The default is not selected.

Show "unused" catalog table columns

Display of catalog table information includes columns marked as "unused" in the *DB2 for z/OS SQL Reference*. This option can be overridden by selecting the **Show all catalog table columns** option.

The default is not selected.

Show "internal use only" catalog table columns

Display of catalog table information includes columns marked as "internal use only" in the *DB2 for z/OS SQL Reference*. This option can be overridden by selecting the **Show all catalog table columns** option.

The default is not selected.

Use automatic column sizing

Varying-length (VARCHAR) columns of the catalog table being displayed may be automatically sized, based on the data stored within the column. Automatic column sizing determines the longest data value in each varying column as the data is read from Db2®. This value is then used to set the display width for the column, prior to the data being displayed.

The default is selected.

Use extended column name headings

Column names for the catalog table being processed are replaced by longer, more meaningful, column headings.

The default is selected.

Interpret columns containing codes

Selected columns in the Db2® catalog table being processed are expanded, based on the value stored within the column. This feature typically operates on one-character columns that contain a code. As an example, the TYPE column in SYSIBM.SYSTABLES is a one-character code containing A,G,M,T,V, or X. The expanded text for these codes is "Alias", "Created global temporary table", "Materialized query table", "Table", "View", and "Auxiliary Table".

This option is useful if you are not familiar with the meanings of the codes in the various Db2® catalog tables. Selecting this option overrides any setting for the **Use extended column name headings** option, and sets that option on.

The default is selected.

Modify order of columns displayed

Columns in the Db2® catalog table may be re-ordered from the order as defined in the Db2® catalog. In addition certain columns may be marked as held. This feature is intended to place the most important columns from the catalog table on the left of the display, in the most logical order. These columns are held, and do not move when scrolling left and right.

The default is selected.

Row count

Limits the number of rows fetched from the Db2® catalog. The value entered applies to:

- Object List selection panels displayed by ZDT/Db2 functions such as the editor, print, copy, import, export, and data create. For more information on object list selection, see Specifying a Db2 object name on page 1063. Note that ZDT/Db2 functions such as edit, copy, print, and so on also have a **Row count** field on the function entry panel. Once an object has been selected, the function entry **Row count** field acts to control the number of rows processed in the specified Db2® object.
- Object lists displayed by these ZDT/Db2 functions:
  - Object List Utility (3.4).
  - Object Privileges Utility (3.5).
  - Explain Utilities (4.5).

A value of 0 or ALL specifies no limit; all rows from the relevant catalog tables are fetched and displayed. When you specify 0, ZDT/Db2 accesses the Db2® catalog using *large mode*, thereby minimizing storage usage for the list of Db2® objects.

If you experience "insufficient storage" conditions, you can specify a value to limit the number of rows fetched from Db2®. Another way to limit the number of rows fetched in the Object List utility (option 3.4) is to specify selection criteria on the function entry panel.

The default is 1000.

Display widths, filtering

Select this option to navigate to the second Object List Options panel. The default is selected. The Object List utility displays information in Db2® catalog tables. Some of the columns within those tables are defined as varying columns with potentially very long maximum lengths.

You can use the **Use automatic column sizing** option to reduce the maximum display width for varying columns to the maximum that is needed to display the data for each column.

Additional control can be achieved by specifying minimum and maximum display widths for the various column types shown on the panel. All values are optional. (See Object List Options (2 of 2) panel on page 746. A minimum value ensures that the display width is at least the value specified, even if there is no displayable data for the column in all rows processed. Specifying a minimum value does not prevent the display width expanding, if needed, to accommodate longer data.

Parent panels

- Set Processing Options panel on page 832

Child panels

To display this panel...	Use/do this
Object List Options (2 of 2) panel on page 746	Press the NxtPage function key (F11)

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- TIME columns on page 168
- INDEXINF primary command on page 970
- "No Primary Key" errors on page 203
- "Row Not Found" errors on page 205

Related references

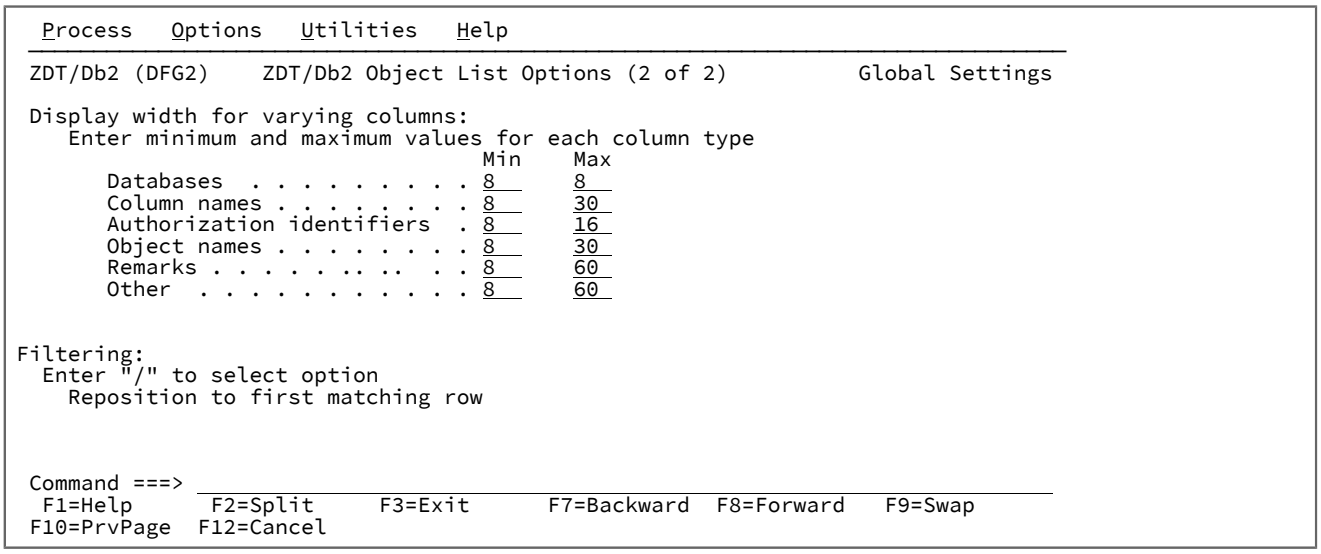
- SQL primary command on page 1005

Object List Options (2 of 2) panel

You use the ZDT/Db2 Object List Options (2 of 2) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 124. ZDT/Db2 Object List Options (2 of 2) panel



Allows you to specify the minimum and maximum display widths for the various column types shown on the object list panel. All values are optional.

Min

Ensures that the display width is at least the value specified, even if there is no displayable data for the column in all rows processed. Specifying a minimum value does not prevent the display width expanding, if needed, to accommodate longer data.

Max

Limits the maximum display width to the value specified, even if data in some rows has a longer width. Specifying a maximum value does not prevent the display width being set to a smaller value, if no rows have data exceeding the value specified.

The following column types are available:

Databases

The name of a Db2® database or table space. The default minimum and maximum values are both 8.

Column names

The name of a Db2® column. The default minimum and maximum values are 8, 30.

Authorization identifier

A Db2® authorization identifier, for example the current SQLID. The default minimum and maximum values are 8, 16.

Object Names

The name of a Db2® object, such as a table, view or alias. The default minimum and maximum values are 8, 30.

Remarks

Applies to columns that are used to store data that is entered by using the COMMENT ON or LABEL ON statements. The default minimum and maximum values are 8, 60.

Other

Applies to any column that does not match the preceding types. The default minimum and maximum values are 8, 60.

Reposition to first matching row

This setting affects how data is displayed after you change the filter values for one or more columns. Use filters to limit the rows that are displayed. The default is not selected.

When you select this setting, the data that is shown after you change the filter values is repositioned to the first matching row.

Parent panels

- Set Processing Options panel on page 832

Child panels

To display this panel…	Use/do this
Object List Options (1 of 2) panel on page 743	Press the PrvPage function key (F10)

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- TIME columns on page 168
- INDEXINF primary command on page 970
- "No Primary Key" errors on page 203
- "Row Not Found" errors on page 205

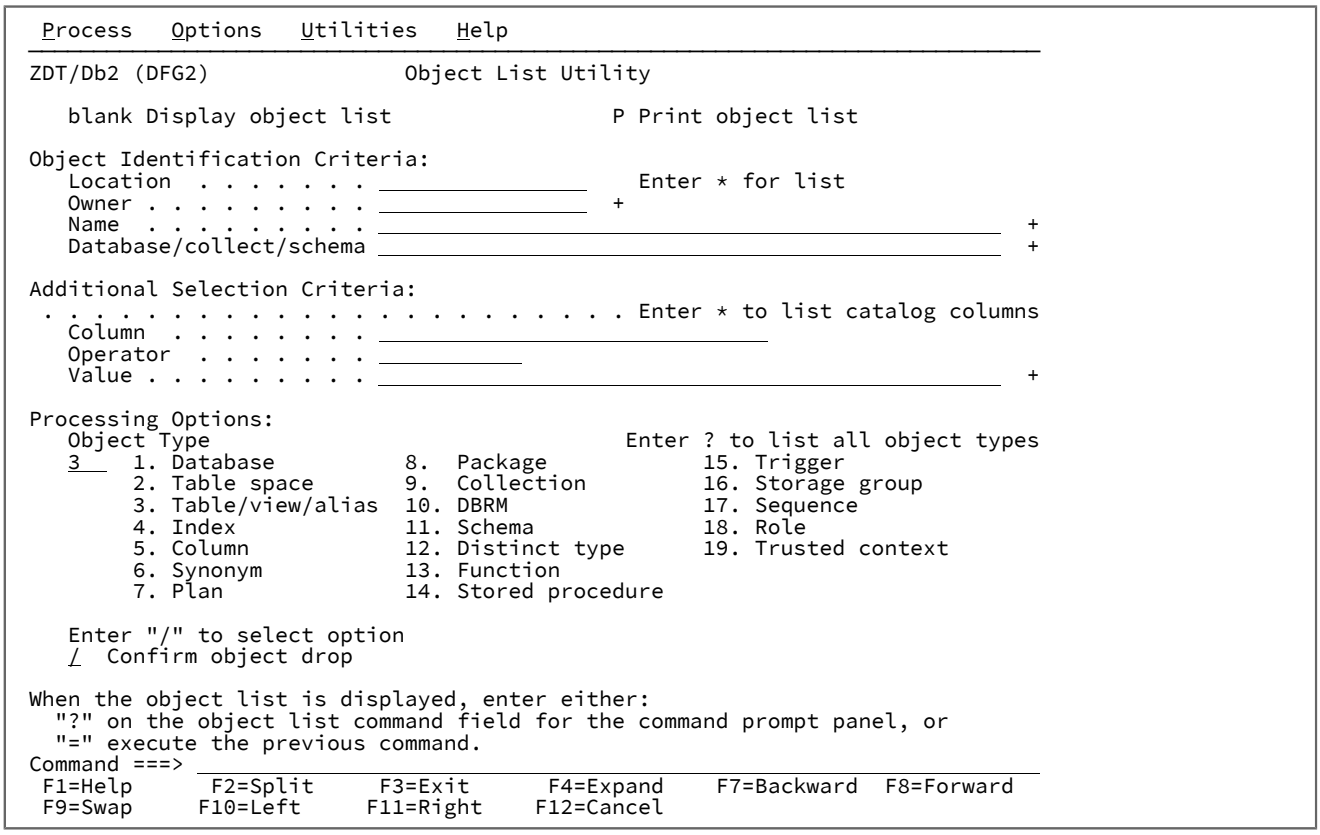
Related references

- SQL primary command on page 1005

Object List Utility panel

You use the Object List Utility panel to specify details about the objects you want to list or print.

Panel and field definitions



Location

The location of the objects you want listed

Owner

The name of the person who created the Db2® object

Name

The name of the Db2® object

Database/collect/schema

The name of a database, a collection, or a schema

Column

Name of column in the relevant Db2® system catalog table

Operator

A valid SQL operator

Value

The value to be compared with the contents of the column

Object Type

The type of Db2® object that you want listed. The default is 3 (**Tables**).

Confirm object drop

Indicates if you want ZDT/Db2 to display a confirmation panel each time you enter a DR(op) command against an object

Parent panels

- Utility Functions panel on page 915

Child panels

To display this panel...	Use/do this
Databases panel on page 609	Object Type 1
Table Spaces panel on page 879	Object Type 2
Tables, Views and Aliases panel on page 891	Object Type 3
Indexes panel on page 723	Object Type 4
Columns panel on page 503	Object Type 5
Synonyms panel on page 855	Object Type 6
Application Plans panel on page 473	Object Type 7
Application Packages panel on page 469	Object Type 8
Collections panel on page 482	Object Type 9
Database Request Modules panel on page 607	Object Type 10
Schemas panel on page 812	Object Type 11
Distinct Types panel on page 637	Object Type 12
Functions panel on page 709	Object Type 13
Stored Procedures panel on page 852	Object Type 14
Triggers panel on page 903	Object Type 15

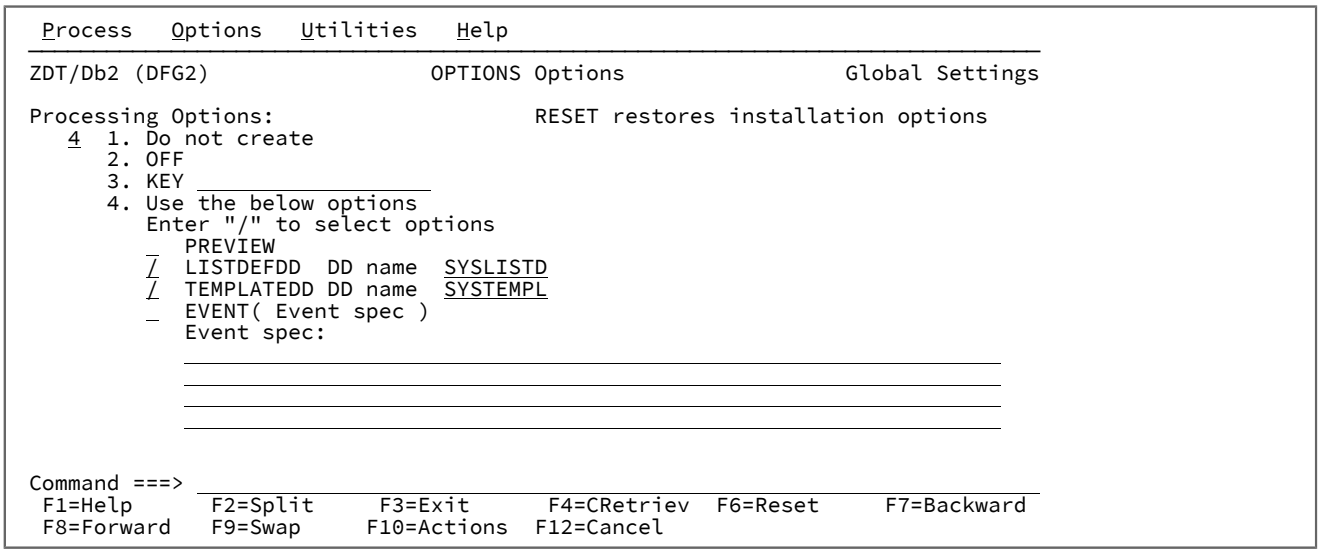
Related tasks

- Working with lists of Db2 objects on page 303

OPTIONS Options panel

You use the OPTIONS Options panel to collect information used in generating an OPTIONS statement in a utility batch job.

Panel and field definitions



Processing Options

The options in capitals are keywords that are used in the OPTIONS statement if the option is selected. The default is 1 (**Do not create**). For more information, see OPTIONS statement in *DB2 Utility Guide and Reference*.

The processing options are:

1. Do not create

When this option is selected, ZDT/Db2 does not create the OPTIONS statement in a utility batch job.

2. OFF

When this option is selected, ZDT/Db2 restores all default options in a utility batch job.

3. KEY

When this option is selected, ZDT/Db2 uses the KEY clause in the OPTIONS statement.

You should only use this option when instructed to do so by HCL Support. Enter the value provided by HCL Support in the **KEY** field. The default for the **KEY** field is blank.

4. Use the below options:

When this option is selected, you can select other keywords for addition to the OPTIONS statement. The relevant keyword is only added to the OPTIONS statement if the option is selected. The options are:

PREVIEW

When this option is selected, the generated utility batch job, when executed, runs in PREVIEW mode. This option corresponds to the PREVIEW keyword in an OPTIONS statement. The default is not selected.

LISTDEFDD

When this option is selected, it specifies the DD name of the LISTDEF definition library. The default DD name for the input field is *"SYSLISTD"*. You can enter a different DD name. The specified DD name appears as a read-only field in the LISTDEF Options panel on page 729.

This option corresponds to the LISTDEFDD clause in an OPTIONS statement. The default is not selected.

TEMPLATEDD

When this option is selected, it specifies the DD name of the TEMPLATE definition library. The default DD name for the input field is *"SYSTEMPL"*. You can enter a different DD name. The specified DD name appears as a read-only field in the TEMPLATE Options (1 of 2) panel on page 897.

This option corresponds to the TEMPLATEDD clause in an OPTIONS statement. The default is not selected.

EVENT

When this option is selected, the utility batch job includes the specified events or events. Enter the events on the following lines. You must enter at least one. The values you enter are placed immediately after the EVENT keyword in the OPTIONS statement, without further validation by ZDT/Db2.

This option corresponds to the EVENT clause in an OPTIONS statement.

The default is not selected. The default for the **Event spec** field is blank.

All fields have installation-defined defaults. You can reset all values to the installation defaults (site policy) at will by using the RESET command or RESET PF Key. Each subsystem has its own installation-defined defaults.

As an example, the following clauses are added to any generated OPTION statement when the options in the figure are selected:

- LISTDEFDD SYSLISTD
- TEMPLATEDD SYSTEMPL

For the Db2® defaults settings of SYSLISTD for LISTDEFDD and SYSTEMPL for TEMPLATEDD, it is also acceptable to *"optimize"*, which means to not generate these clauses, since they are the Db2® defaults.

Child panels

None.

Related tasks

- RESET primary command on page 994
- Setting options for Db2 utilities on page 348
- Db2 Utility OPTIONS options (option 3.UO) on page 70

Package Dependencies panel

ZDT/Db2 displays the Package Dependencies panel if you use the DEP line command against an item listed on the Application Packages panel on page 469.

Child panels

None.

Related tasks

Package List panel

ZDT/Db2 displays the Package List panel if you use the PKL line command against an item listed on:

- The Application Plans panel on page 473, or
- The Application Packages panel on page 469, or
- The Collections panel on page 482

Related tasks

Parents of Table panel

ZDT/Db2 displays the Parents of Table panel if you use the PAR line command against an item listed on the Tables, Views and Aliases panel on page 891.

Partitioned Table Statistics panel

ZDT/Db2 displays the Partitioned Table Statistics panel if you use the PST line command against an item listed on the Tables, Views and Aliases panel on page 891.

Plan Dependencies panel

ZDT/Db2 displays the Plan Dependencies panel if you use the DEP line command against an item listed on the Application Plans panel on page 473.

Plan Enabled/Disabled Connections panel

ZDT/Db2 displays the Plan Enabled/Disabled Connections panel if you use the EDC line command against an item listed on the Application Plans panel on page 473.

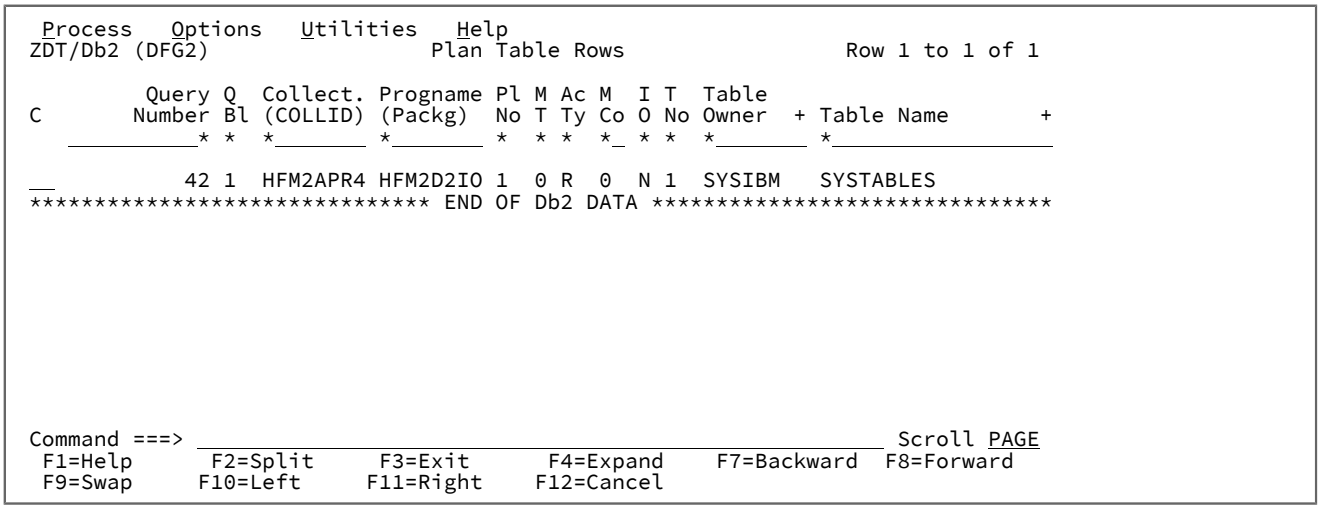
Plan Table Rows panel

ZDT/Db2 displays the Plan Table Rows panel if you:

- Select option **L** (List plan table rows) on the Explain Utilities panel, or
- Enter the EXPLAIN primary command on the Enter, Execute and Explain SQL Statements with the **Show results** option selected.

Use this panel to display rows from a plan table.

Figure 125. Plan Table Rows panel (collection ID format)



The Plan Table Rows panel shows you rows from a plan table:

C

Input field where you can enter one of the following line commands:

D

Delete row

DK

Delete rows for package

DP

Delete rows for plan

DQ

Delete rows for query number

FR

Show function table rows. ZDT/Db2 displays the Function Table Rows panel (not shown).

I

Interpret EXPLAIN data

IH

Insert optimizer hint

K

Show package

M

Show DBRM

P

Show plan

SR

Show statement table rows. ZDT/Db2 displays the Statement Table Rows panel (see Statement Table Rows panel on page 845).

T

Show table

UH

Update optimizer hint

X

Show index

Query Number

A number that identifies the SQL statement.

Q Bl

Query block number: the number reflects the query's order of appearance in the EXPLAIN statement.

Collect.

Collection ID or blank for dynamic EXPLAIN. This information only is displayed for the COL format of the panel.

Applname

PLAN name or blank for dynamic EXPLAIN. This information only is displayed for the PLAN format of the panel.

Hint ID

Optimization hint ID or blank for dynamic EXPLAIN. This information only is displayed for the HINT format of the panel.

Progname

DBRM/package name. This information only is displayed for the COL and PLAN formats of the panel.

Hint Used

The optimization hint used or blank for dynamic EXPLAIN. This information only is displayed for the HINT format of the panel.

Pl No

Plan number. Indicates order in which the statement is run.

M T

Method: Join method that is used.

Ac Ty

Access type. Method of accessing the table.

M Co

Matching columns. Number of index keys used in an index scan.

I O

Index only. Whether only the index is accessed in this step.

T No

Table number. Position of the table in the statement.

Table Owner

Owner of the table being accessed.

Table Name

Name of the table being accessed.

Parent panels

- Explain Utilities panel on page 686

Child panels

To display this panel...	Use/do this
EXPLAIN Interpretation panel on page 686	Line command I
Set Optimizer Hint panel on page 823	Line commands IH, UH
Application Packages panel on page 469	Line command K
Statement Table Rows panel on page 845	Line command SR
	Line command T

To display this panel...	Use/do this
Tables, Views and Aliases panel on page 891	

Related tasks

- Explaining SQL on page 427
- Managing tables used by SQL explain on page 430

Related references

- COL primary command on page 934
- PLAN primary command on page 986
- HINT primary command on page 968

Primary Key for Table panel

ZDT/Db2 displays the Primary Key for Table panel if you use the PK line command against an item listed on the Tables, Views and Aliases panel on page 891.

Primary Option Menu panel

You use the Primary Option Menu panel to select the ZDT/Db2 function you want to perform.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DF72)              Primary Option Menu

0  Settings      Set processing options              User ID . : SHRIKE2
1  View          View Db2 object                     System ID : FMD2
2  Edit          Edit Db2 table                      Appl ID . : HFM2
3  Utilities     Perform utility functions           Version . : 17.1.0
4  SQL           Prototype, execute and analyze SQL  Terminal  : 3278
5  Db2I          Start Db2 Interactive               Screen  . : 1
6  Command       Enter and execute a Db2 Command     Date  . . : 2018/11/04
7  Template      Template utilities                  Time  . . : 11:22
X  Exit          Terminate ZDT/Db2
                                                     Db2 SSID  . DF72
                                                     SQL ID  . . SHRIKE2




 Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Parent panels

- (z/OS® Primary Option Menu)

Child panels

To display this panel…	Use/do this
Set Processing Options panel on page 832	Option 0
Db2 View panel on page 631	Option 1
Db2 Edit panel on page 616	Option 2
Utility Functions panel on page 915	Option 3
SQL Prototyping, Execution and Analysis panel on page 843	Option 4
Db2I Primary Option Menu panel on page 635	Option 5
Enter and Execute Db2 Commands panel on page 683	Option 6
Db2 Browse panel on page 612	Enter "B" on command line

Related tasks

- Viewing and changing Db2 data on page 122

Print Audit Trail panel

You use the Print Audit Trail panel to specify the name of the Audit Trail data set to be used to generate an Audit Trail report.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                Print Audit Trail

Audit Trail:
   Data set name . . . _____________________________________________
   Description . . . . ____________________________________

Processing Options:
   Enter "/" to select option           Formatted display options
   /  Print only changed columns         2  1. Hex Format
   _  Keep data set after printing          2. Non-display Hex
   /  Browse report                      /  Highlight changes
   _  Batch execution




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Data set name

The name of the Audit Trail data set to be used.

Description

A description for the report. If the descriptions contains blanks, you must enclose it in quotation marks. This field is optional.

You can select the following options by entering a "/" in the selection field:

Print only changed columns

Limits the amount of printed lines.

Keep data set after printing

Whether to keep or delete the data set after it is printed.

Browse report

The audit trail report is displayed using Print Browse (option 3.11).

Batch execution

Generates JCL for printing the audit trail report using a batch job. For command syntax, see the AUD function in Z Data Tools User's Guide and Reference.

Hex Format

To produce an UPDOWN hexadecimal display below the standard field display.

Non-display Hex

To produce an UPDOWN hexadecimal display below the standard field display only for fields that contain non-displayable \ characters.

Highlight changes

Highlight the changed fields. An asterisk is placed to left of the before data to indicate the field has been changed.

Parent panels

- Utility Functions panel on page 915

Child panels

None.

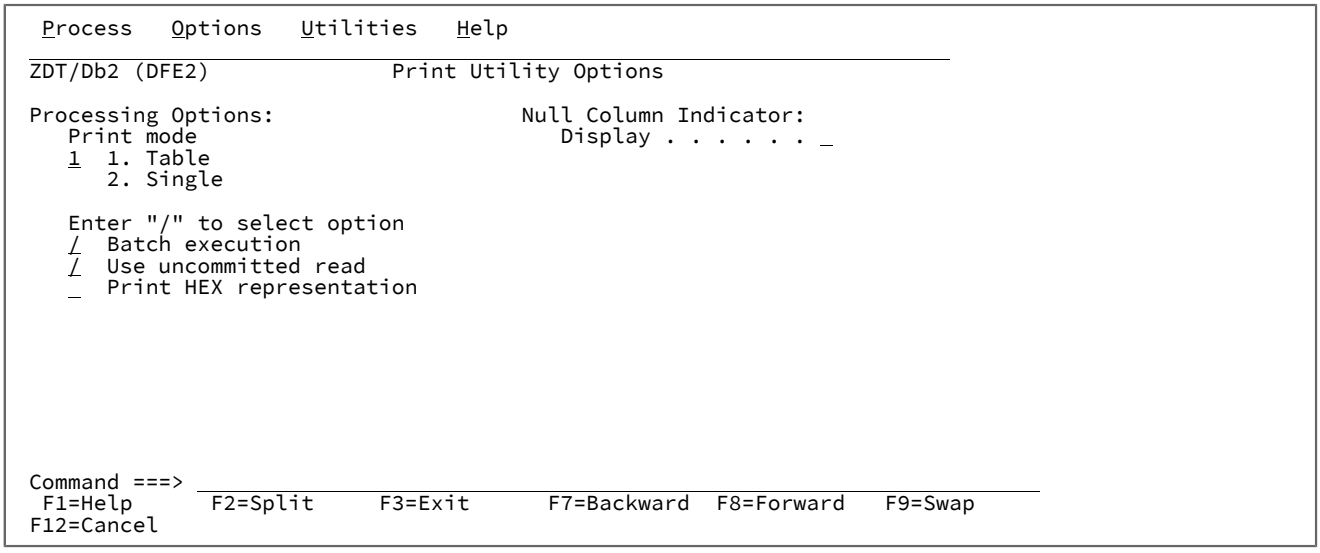
Related tasks

- Printing an audit trail report on page 335

Print Utility Options panel

You use the Print Utility Options panel to set various options for the Print utility.

Panel and field definitions



Print Mode

The format of the print output. The default is 1 (**Table**):

1. Table

In Table print mode (the default mode), the rows of the table are formatted into columns and printed horizontally across the page. Each column has a heading of either the column name or a heading specified by you. Multiple rows are printed on each page. Each column of the table is formatted according to the data type for that column. For example, numeric data types are printed as whole or decimal numbers, and date and time columns are printed as dates or times. Print data might be truncated, depending on the LRECL of the print data set.

2. Single

In Single mode, only one column of the table is printed on each line of the page. The columns of the table are arranged vertically on the page, with the column header on the left of the page, and the data for that column on the right. Each column of the table is formatted according to the data type for that column. For example, numeric data types are printed as whole or decimal numbers, and date and time columns are printed as dates or times.

Display

Null Column Display Indicator. This option, applying to columns that allow Db2® nulls only, allows you to specify a character to represent the Db2® null value for the ZDT/Db2 print utility. The default is the underscore character.

Batch execution

Builds the necessary JCL to perform the Print function in a batch job. The generated JCL is displayed using the ISPF editor, where you can review the JCL and submit the job for batch processing. The default is selected, that is, the print is executed in batch.

Use uncommitted read

Enables data to be read that has not yet been committed. The default setting is selected; that is, ZDT/Db2 uses uncommitted read when retrieving the data from Db2®.

Note: If ZDT/Db2 attempts to retrieve data that has a write lock, an SQL error might result.

Print HEX representation

Prints the hexadecimal representation of the data. The default is not selected; that is, the data is not printed in hexadecimal format.

Parent panels

- Set Processing Options panel on page 832
- Print Utility panel on page 764

Child panels

None.

Related tasks

- Printing the contents of a Db2 table on page 328
- Selecting options on ZDT/Db2 panels on page 45

Print Utility panel

You use the Print Utility panel to print the contents of a table or view.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)                  Print Utility

Db2 Object:
   Location  . . . . . _________________        Database . . _________ (optional)
   Owner . . . . . . . SYSIBM           +     Table space  _________ (optional)
   Name  . . . . . . . SYSCOPY                                          +

   Row count . . . . . 20_____       Number of rows to print

Db2 Template:
   Data set name . . . _____________________________________________
   Member  . . . . . . ________

Processing Options:
   Print Mode       Template usage              Enter "/", "A"lways to select
   3  1. Above                          _  Edit options
      2. Previous                       _  Edit template
      3. Generate from table            _  Create audit trail (*)
      4. Generate/Replace
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Location

The location of the Db2® object. Leave this field blank if the table or view that you want to work with is at your current location. This field is optional.

database

The name of the database to which you want to restrict your selection. This field is optional.

Owner

The authorization ID of the owner of the Db2® object. This field is optional.

Table space

The name of the table space to which you want to restrict your selection. This field is optional.

Name

The name of the Db2® object.

Row count

The number of rows of the Db2® object you want to print.

Valid values:

0

All rows printed

1–99999999

Number of rows printed

ALL

All rows printed. This is the default.

Data set name

The name of the template data set. It can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Processing Options

You can use these options to specify, amongst other things, the format in which you want the data printed, which template you want to use for the print process, and whether you want to edit the template before the print process.

Template usage

The Print Utility panel provides four options for template processing:

1. **Above**

   Requires that you enter the name of a template data set (and optionally a member name) in the **Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. After you have specified the member containing a template, processing continues using this template in place of the automatically-generated template for the table (for a detailed explanation, see Template Usage option 3).

2. **Previous**

   Uses the last (previously used) template for this table.

3. **Generate from table**

   Generates a template based on the information in the Db2® catalog for the specified table. This is the default setting. If you have specified a member name in the **Template** section of the panel, it is ignored. ZDT/Db2 generates the required template in memory.

4. **Generate/Replace**

   Generates a template based on the information in the Db2® catalog for the specified table, and saves this information in the member specified in the **Template** section of the panel. If the member already exists, it is replaced. Any prior customization is lost.

You can select the following options by entering a "/" or an "A":

Edit options

Use to display the ZDT/Db2 print utility options, prior to printing the data for the Db2® object. The default is not selected.

Edit template

Use to edit the template before printing the data. The default is not selected.

Create audit trail

Controls whether access to Db2® by ZDT/Db2 is written to the ZDT/Db2 audit log. This option only appears when SAF-rule controlled auditing is in effect.

ZDT/Db2 may write audit records to either SMF, or a data set.

When SAF rules are used to control ZDT/Db2 auditing, the decision to write audit records is based on the Db2® object name being processed, and the current userid (TSO logonid). Therefore auditing may be required when processing one Db2® object, but not another. In addition, it is possible that the auditing requirement cannot be changed for a particular Db2® object; this might be different for another Db2® object, and the requirement may be different for other TSO userids.

If you do not have the authority to change the audit setting when processing a particular Db2® object, ZDT/Db2 prompts you if you enter the wrong setting. If you do have the authority to set the audit setting for a particular Db2® object, ZDT/Db2 honors whatever value is specified in this field.

Parent panels

- Utility Functions panel on page 915
- Tables, Views and Aliases panel on page 891

Child panels

To display this panel…	Use/do this
Print Utility Options panel on page 763	Select **Edit options**
Column Selection/Edit panel on page 497	Select **Edit template**

Related tasks

- Printing the contents of a Db2 table on page 328
- Selecting options on ZDT/Db2 panels on page 45
- Working with templates on page 74
- SAF-rule controlled auditing on page 71

Privilege details panels

These panels are:

- Application Plan Privilege Details panel
- Column Privilege Details panel
- Database Privilege Details panel
- Distinct Type Usage Privilege Details panel
- Function Privilege Details panel
- Package Privilege Details panel
- Schema Privilege Details panel
- Stored Procedure Privilege Details panel
- Table Space Use Privilege Details panel
- Table/View Privilege Details panel

ZDT/Db2 displays one of these panels when you specify the I line command against an object displayed in a privilege list of Db2® objects.

Privileges panels

These panels are:

- Application Plan Privileges panel
- Collection Privileges panel
- Column Privileges panel
- Database Privileges panel
- Distinct Type Privileges panel
- Function Privileges panel
- Package Privileges panel
- Schema Privileges panel
- Stored Procedure Privileges panel
- Table Privileges panel
- Table Space Use Privileges panel

ZDT/Db2 displays one of these panels if you use the Privileges utility function (3.5) to display privilege details for an object type, or if you specify the P line command against an object displayed in a list of Db2® objects.

Format of privilege display panels

All of the privilege display panels use an ISPF table display of information retrieved from the Db2® catalog. You can scroll through the information in a typical ISPF manner. The information is presented in read-only format.

For each privilege display panel:

- You can use the **Cmd** field to enter a line command. The line commands you can enter vary according to the object type. To display a list of valid line commands for the type of objects you are viewing, enter a "?" in the **Cmd** field.
- Single-character fields show the current status for each privilege that can apply to the object type listed:

  Blank

  The grantee does not have the specified privilege.

  Y

  The grantee has the specified privilege, but cannot grant this privilege to other users.

  G

  The grantee has the specified privilege, and can grant this privilege to other users.
- At the top of each column of data, immediately below the column name, there are input fields. When the panel is initially displayed, all of these input fields contain an asterisk (*) indicating "all values". You can use one or more of these input fields to restrict the list of objects listed. For example, by specifying `BRA` in the entry field below the **Grantor** column name on the Database Privileges panel shown in Privileges panels on page 768, the list of databases is restricted to those where the name of the grantor starts with "BRA".

Related tasks

- Displaying privileges on page 342
- Working with lists of Db2 objects on page 303

Rebind Application Plan panel

Panel and field definitions

Figure 126. Rebind Application Plan panel (1 of 2)

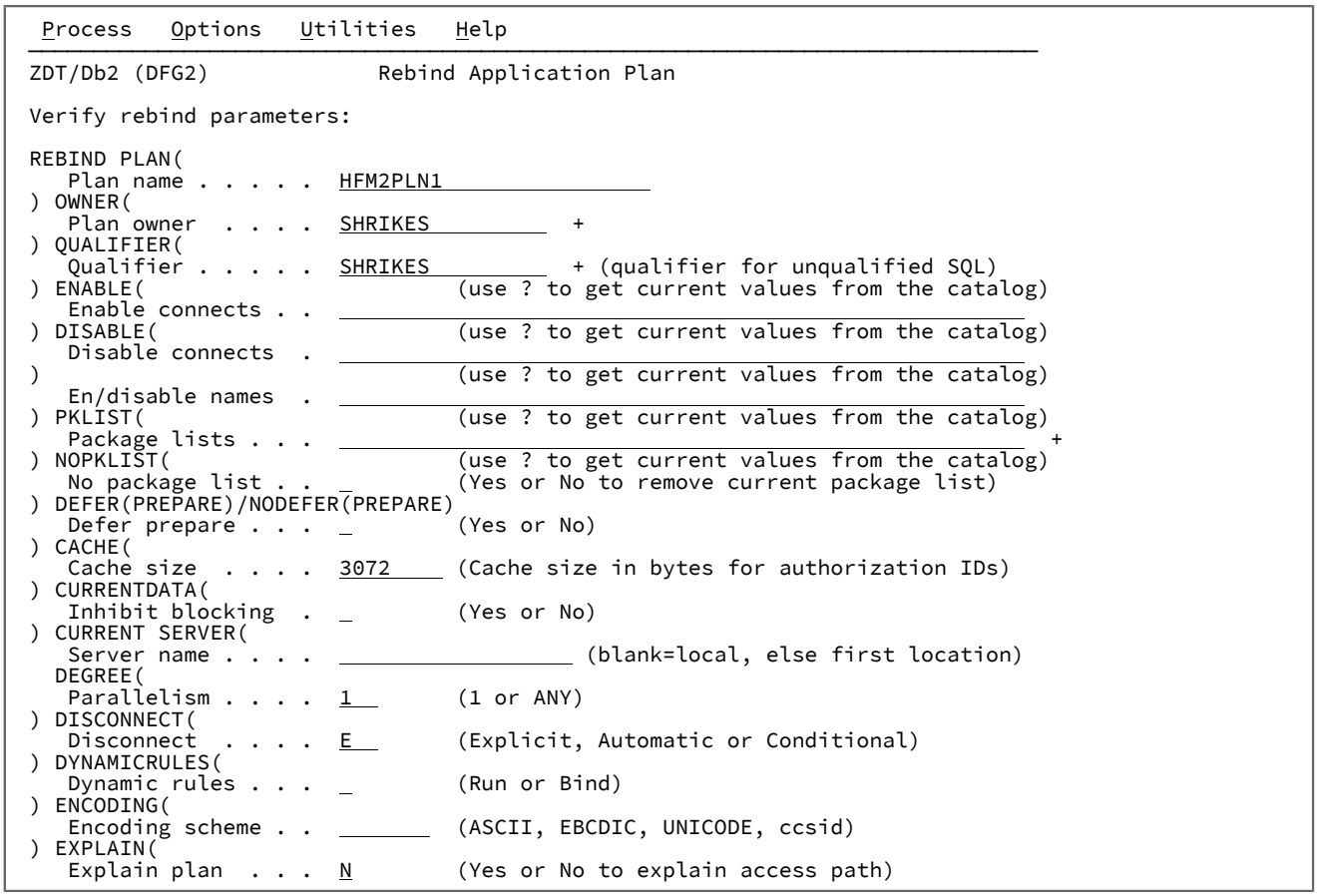
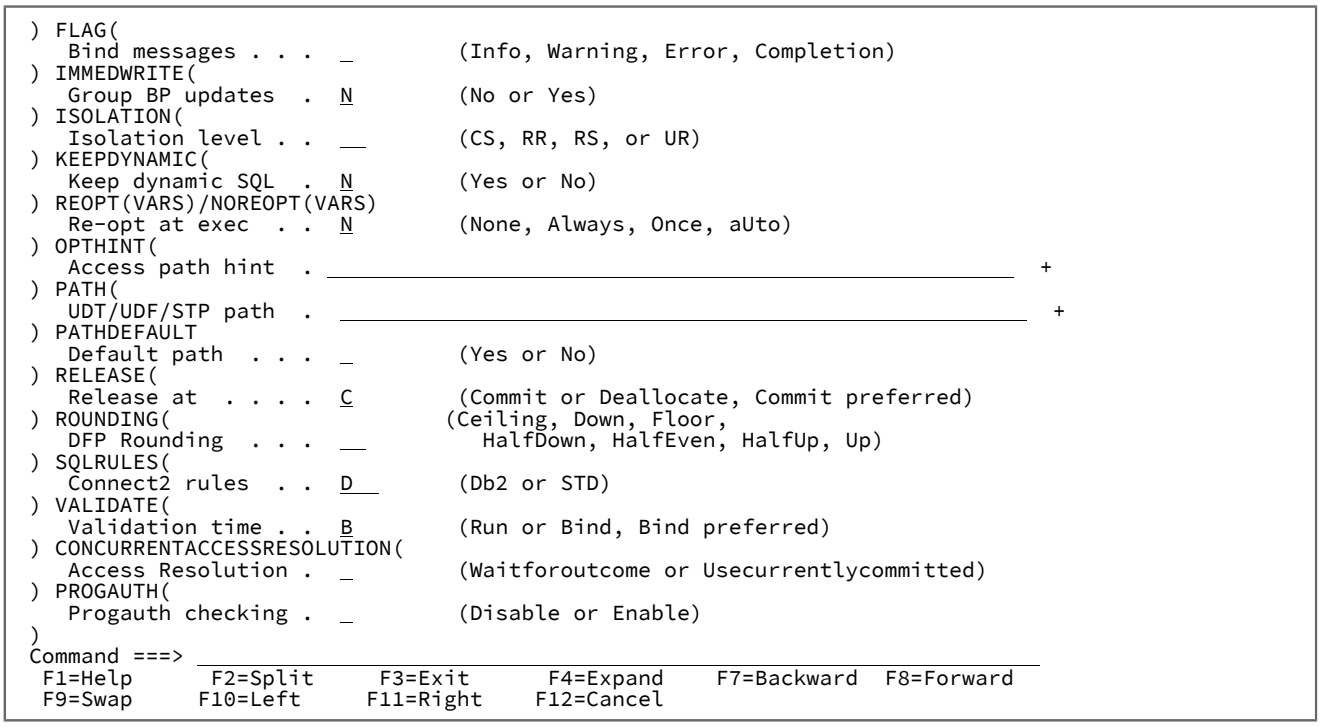


Figure 127. Rebind Application Plan panel (2 of 2)



Parent panels

- Application Plans panel on page 473

Child panels

None.

Related tasks

Rebind Package panel

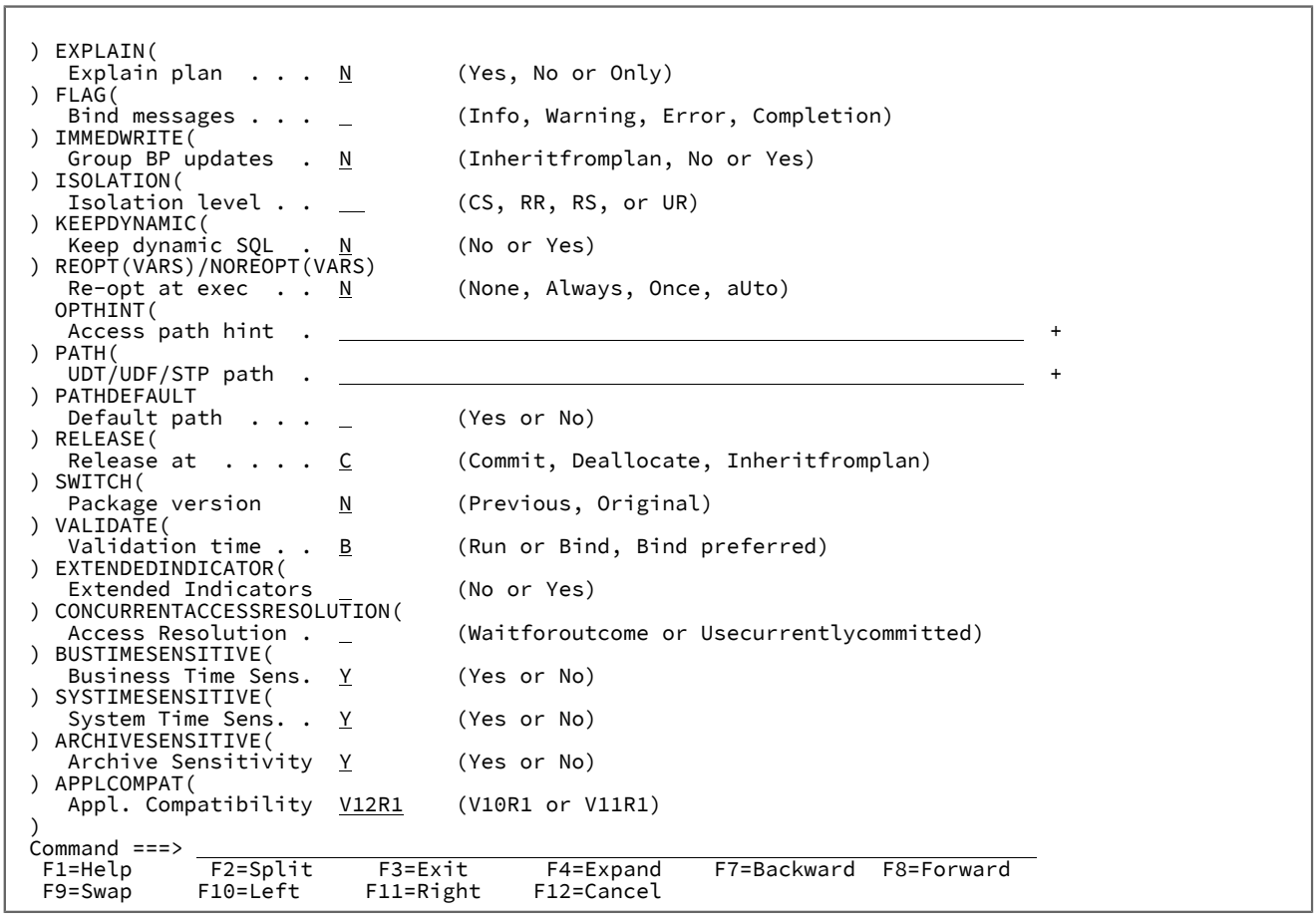
Panel and field definitions

Figure 128. Rebind Package panel (1 of 2)

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Rebind Package
                                                                More:     +

Verify rebind parameters:
REBIND PACKAGE(
   Location  . . . . . _______________
   Collection  . . . . DSNUTIL                                              +
   Package . . . . . . DSNUGSQL          +    (
   Version . . . . . . ______________________________________________________ +
)) OWNER(
   Package Owner . . . ARROWSB           +
) QUALIFIER(
   Qualifier . . . . . ARROWSB           +
) ENABLE(                          (use ? to get current values from the catalog)
   Enable connects . . ______________________________________________________ +
) DISABLE(                         (use ? to get current values from the catalog)
   Disable connects  . ______________________________________________________ +
)                                  (use ? to get current values from the catalog)
   En/disable names  . ______________________________________________________ +
  PLANMGMT(
   Plan management . . _            (Off, Basic, Extended or blank)
) APREUSE(
   Reuse access paths  N            (None, Error or Warn)
) APREUSESOURCE(
   Package copy  . . . _            (Current, Previous or Original)
) APCOMPARE(
   Comp access paths . _            (None, Error or Warn)
) APRETAINDUP(
   Retain old copies . Y            (Yes or No)
) GETACCELARCHIVE(
   Archived Accel data _            (No or Yes)
) QUERYACCELERATION(               (None, All, Enable,
   Acceleration option _              enableWithfailback, eliGible)
) CONCENTRATESTMT(
   Concentrate stmt. . N            (No or Yes)
) CURRENTDATA(
   Inhibit blocking  . N            (Yes or No)
) DBPROTOCOL(
   3 part protocol . . D            (DRDA)
) DEFER(PREPARE) / DEFER(INHERITFROMPLAN) / NODEFER(PREPARE)
   Defer prepare . . . _            (Yes, Inherit or No)
  DEGREE(
   Parallelism . . . . 1__          (1 or ANY)
) DESCSTAT(
   Describe SQL area . Y            (No or Yes)
) DYNAMICRULES(
   Dynamic rules . . . __           (Run, Bind, Define, Invoke)
) ENCODING(
   Encoding scheme . . EBCDIC_      (ASCII, EBCDIC, UNICODE, ccsid)
```

Figure 129. Rebind Package panel (2 of 2)



Parent panels

- Application Packages panel on page 469

Child panels

None.

Related tasks

REBUILD (Indexes for Table Spaces) panel

You use the REBUILD (Indexes for Table Spaces) panel to generate utility control statements using the REBUILD INDEX (ALL) TABLESPACE form of REBUILD utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)        REBUILD (Indexes for Table Spaces)    Row 1 to 12 of 133


S P                                                    Sort
e R          Table    Part'n                WORKDDN  Device   Sort   Sort
l C Database Space    Number Reuse WORKDDN  Template Type     Num    Keys
_   DSN8D81A DSN8S81D      0   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81E      1   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81E      2   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81E      3   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81E      4   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81E      5   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81P      0   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81R      0   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81A DSN8S81S      0   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81E DSN8S81U      0   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81P DSN8S81C      0   N    SYSUT1     N     SYSALLDA 4       Y
_   DSN8D81P DSN8S81Q      0   N    SYSUT1     N     SYSALLDA 4       Y

Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Sel

Selection column. To select a table space, type S in this column. Selected table spaces show an asterisk (*) in the **PRC** column.

PRC

Processing indicator. An asterisk (*) indicates the item is selected.

Database

The name of the database in which the table space is defined.

Table Space

The name of the table space.

Part'n Number

The partition number for the table space. This is 0 for a non-partitioned table space. For a partitioned table space, the number represents the partition of the table space. There is one row in the table for each partition of a partitioned table space. This is a display-only field.

Reuse

Controls whether Db2® deletes and redefines the index space data set as part of the rebuild operation:

YES

Keeps the existing data set.

NO

Deletes and redefines the index space data set.

WORKDDN

The ddname for the optional temporary work file used by the utility job. The value displayed is either SYSUT1, or another value if you have changed the default ddname. You can type a new value.

WORKDDN Template

Indicates whether the value of the associated **WORKDDN** field is a template name.

Sort Device Type

The device type used by the utility to allocate sort data sets. The value displayed is either SYSALLDA, or another value if you has changed the default device type. You can type a new value.

Sort Num

The number of sort data sets used by the utility. The value displayed is either 4, or another value if you have changed the default setting. You can type a new value.

Sort Keys

Controls whether the REBUILD utility rebuilds indexes in parallel:

YES

Builds the indexes in parallel.

NO

Does not build the indexes in parallel.

You can type a new value.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

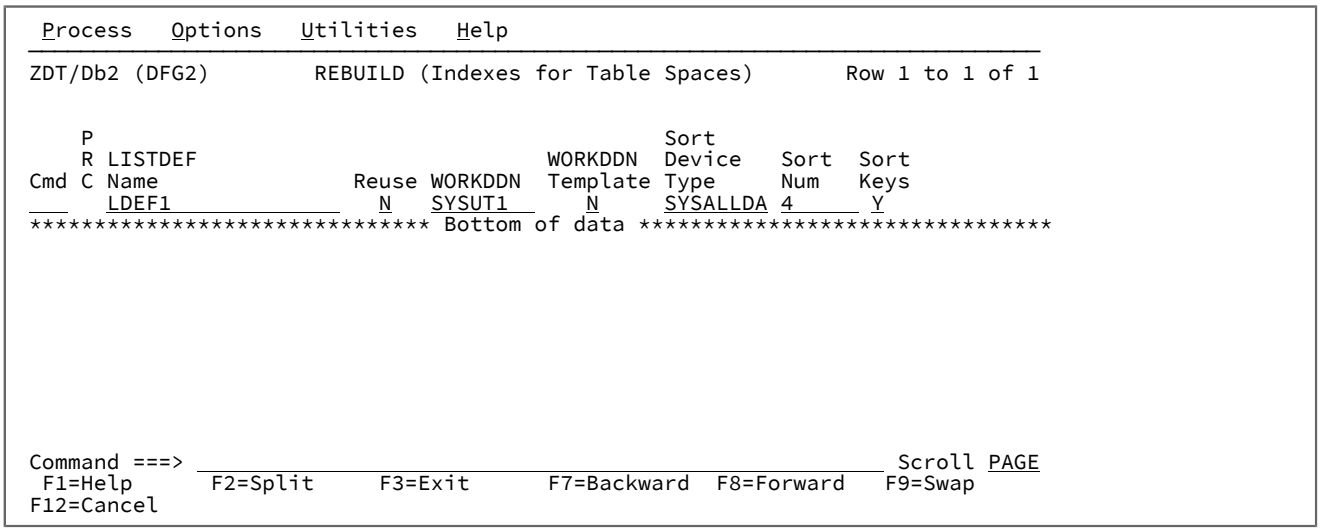
Related tasks

- REBUILD utility on page 365
- REBUILD utility (indexes for table spaces) on page 367
- Loading data described by a copybook or template on page 361

REBUILD (Indexes for Table Spaces) with LISTDEF panel

You use the REBUILD (Indexes for Table Spaces) with LISTDEF panel to generate utility control statements using the REBUILD INDEX TABLESPACE form of the REBUILD utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- REBUILD utility on page 365
- REBUILD utility (indexes for table spaces) on page 367

REBUILD Utility (Indexes) panel

You use the REBUILD Utility (Indexes) panel to generate utility control statements using the REBUILD INDEX (index_name) form of the REBUILD utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)            REBUILD Utility (Indexes)            Row 1 to 1 of 1


    P                                        Sort
    R LISTDEF                    WORKDDN     Device    Sort   Sort
Cmd C Name                Reuse  WORKDDN     Template  Type      Num   Keys
___   LDEFNAME1_________   N     SYSUT1__      N       SYSALLDA  4_____ Y
****************************** Bottom of data ********************************









Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Cmd

Selection column. To select an index, type S in this column. Selected indexes show an asterisk (*) in the **PRC** column.

PRC

Processing indicator. An asterisk (*) indicates the item is selected.

Index Owner

The name of the owner of the index.

Index Name

The name of the index.

Part'n Number

The partition number for the index. This is 0 for a non-partitioned index. For a partitioned index, the number represents the partition of the index. There is one row in the table for each partition of a partitioned index. This is a display-only field.

Reuse

Controls whether Db2® deletes and redefines the index data set as part of the rebuild operation:

YES

Keeps the existing data set.

NO

Deletes and redefines the index data set.

WORKDDN

The ddname for the optional temporary work file used by the utility job. The value displayed is either SYSUT1, or another value if you have changed the default ddname. You can type a new value.

WORK DDN Template

Indicates whether the value of the associated **WORKDDN** field is a template name.

Sort Device Type

The device type used by the utility to allocate sort data sets. The value displayed is either SYSALLDA, or another value if you have changed the default device type. You can type a new value.

Sort Num

The number of sort data sets used by the utility. The value displayed is either 4, or another value if you have changed the default setting. You can type a new value.

Sort Keys

Controls whether the utility rebuilds indexes in parallel:

YES

Builds the indexes in parallel.

NO

Does not build the indexes in parallel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

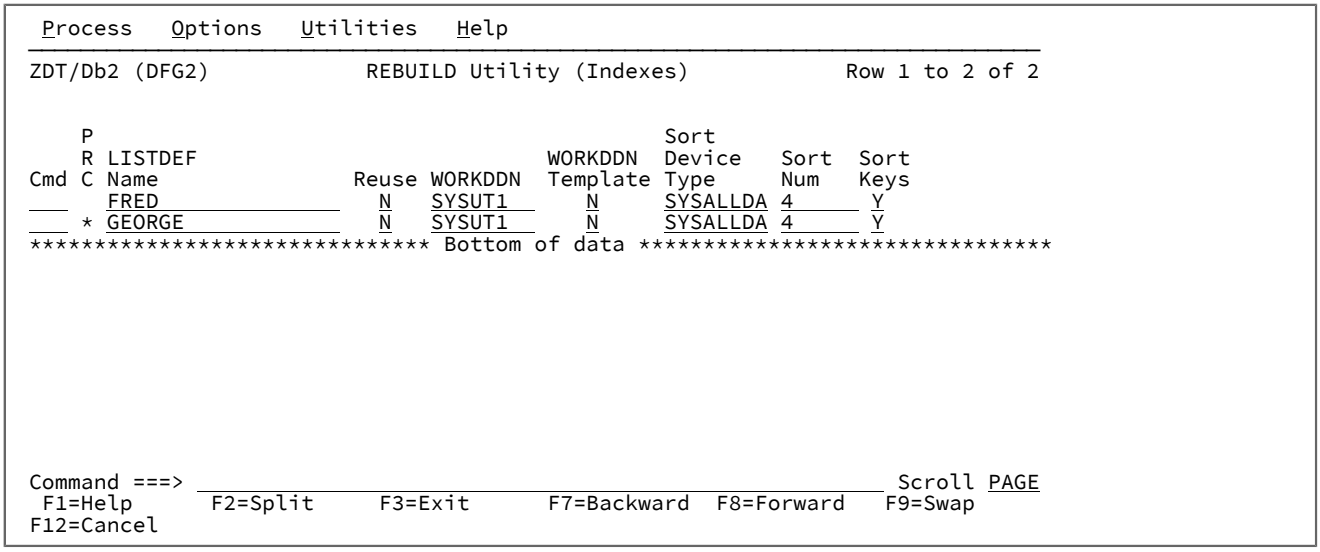
Related tasks

- REBUILD utility on page 365
- REBUILD utility (indexes) on page 366
- Loading data described by a copybook or template on page 361

REBUILD (Indexes) with LISTDEF panel

You use the REBUILD (Indexes) with LISTDEF panel to generates utility control statements using the REBUILD INDEX TABLESPACE form of the REBUILD utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

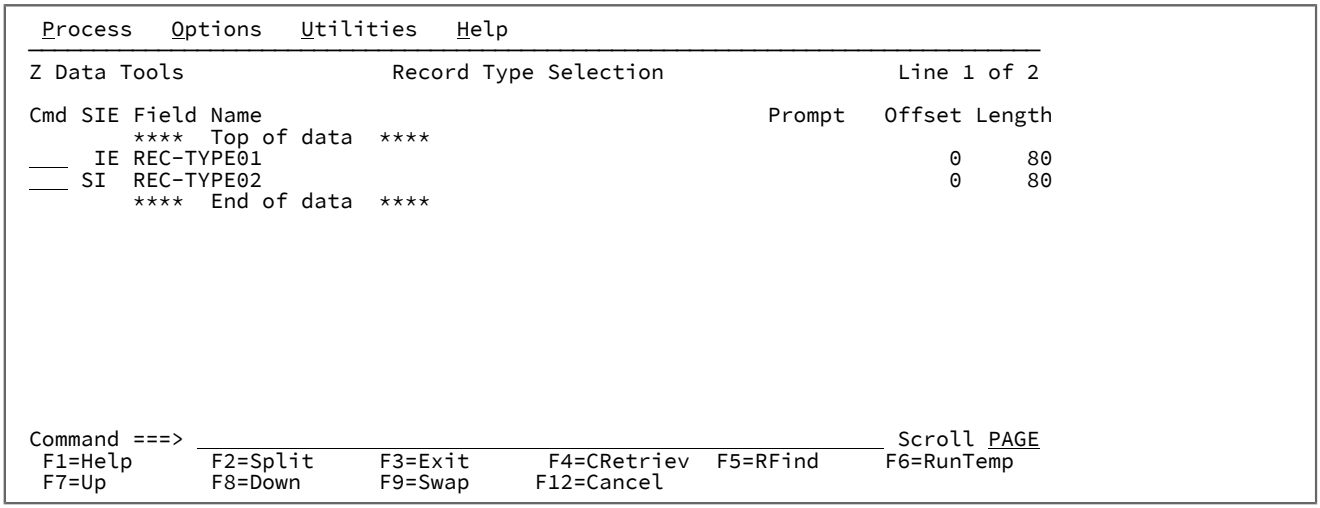
None.

Related tasks

- REBUILD utility on page 365
- REBUILD utility (indexes) on page 366

Record Type Selection panel

Panel and field definitions



Parent panels

- LOAD Utility - Using Templates panel on page 735

Child panels

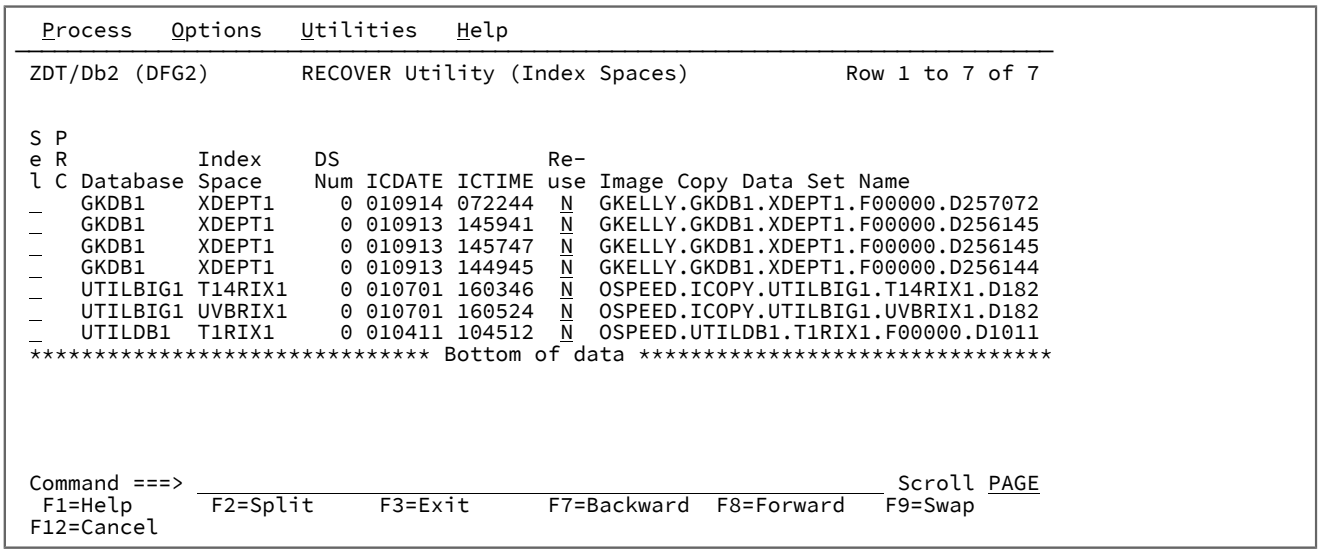
None.

Related tasks

RECOVER Utility (Index Spaces) panel

You use the RECOVER Utility (Index Spaces) panel to generate utility control statements using the RECOVER INDEXSPACE form of the RECOVER utility. You can only copy and recover indexes that have been created with the COPY YES clause.

Panel and field definitions



The fields in the ISPF table part of the panel are:

Sel

Selection column. Type s to select an index.

PRC

Processing column. An asterisk (*) indicates the selected index spaces.

Database

The name of the database in which the index space is defined.

Index Space

The name of the index space.

DS Num

The partition number for this table space. Non-partitioned table spaces have a partition number of 0. For a partitioned table space, the number indicates the partition of the table space.

ICDATE

The date that the image copy was taken. The format is *yymmdd*.

ICTIME

The time that the image copy was taken. The format is *hhmmss*.

Reuse

Reuse option:

Y

The data set for the index space is not deleted and is re-defined as part of the RECOVER utility processing.

N

The data set for the index space is deleted and redefined as part of the RECOVER utility processing.

Image copy data set name

The name (which must be cataloged) of the image copy data set. Note that the existence of an entry in the catalog table does not mean that the data set still exists.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- RECOVER (index spaces) on page 370

RECOVER Utility (Index Spaces) with LISTDEF panel

You use the RECOVER Utility (Index Spaces) with LISTDEF panel to generate utility control statements using the RECOVER INDEXSPACE form of the RECOVER utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)          RECOVER Utility (Index Spaces)         Row 1 to 1 of 1


    P
    R LISTDEF
Cmd C Name                 Reuse
___   LDEF1                  N
****************************** Bottom of data ********************************









Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

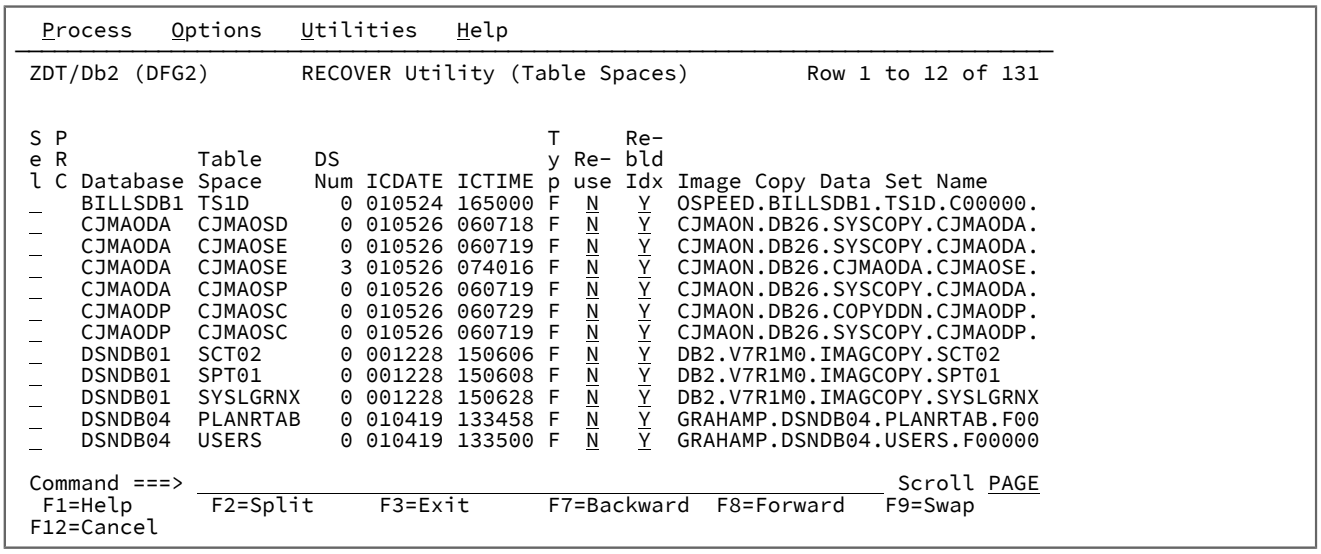
Related tasks

- RECOVER utility on page 368
- RECOVER (index spaces) on page 370

RECOVER Utility (Table Spaces) panel

You use the RECOVER Utility (Table Spaces) panel to generate utility control statements using the RECOVER TABLESPACE form of the RECOVER utility.

Panel and field definitions



The fields in the ISPF table part of the panel are:

Sel

Selection column. Type s to select a table space.

PRC

Processing column. An asterisk (*) indicates the selected table spaces.

Database

The name of the database in which the table space is defined.

Table Space

The name of the table space.

DS Num

The partition number for this table space. Non-partitioned table spaces have a part number of 0. For a partitioned table space, the number indicates the partition of the table space.

ICDATE

The date that the image copy was taken. The format is *yymmdd*.

ICTIME

The time that the image copy was taken. The format is *hhmmss*.

Typ

The type of entry found in the SYSIBM.SYSCOPY table:

F

Full image copy.

I

Incremental image copy.

Reuse

Reuse option:

Y

The data set for the table space is not deleted and re-defined as part of the RECOVER utility processing.

N

The data set for the table space is deleted and redefined as part of the RECOVER utility processing.

Re-bld Idx

Rebuild indexes after recovery:

Y

A REBUILD INDEX (ALL) TABLE SPACE command is included in the JCL to perform a full rebuild of all of the indexes on all of the tables in the table space. This is the standard method for recovering a table space, and prevents the indexes being left in an unusable rebuild pending (RBDP) state.

N

Indexes not rebuilt.

Image copy data set name

The name (which must be cataloged) of the image copy data set. Note that the existence of an entry in the catalog table does not mean that the data set (still) exists.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

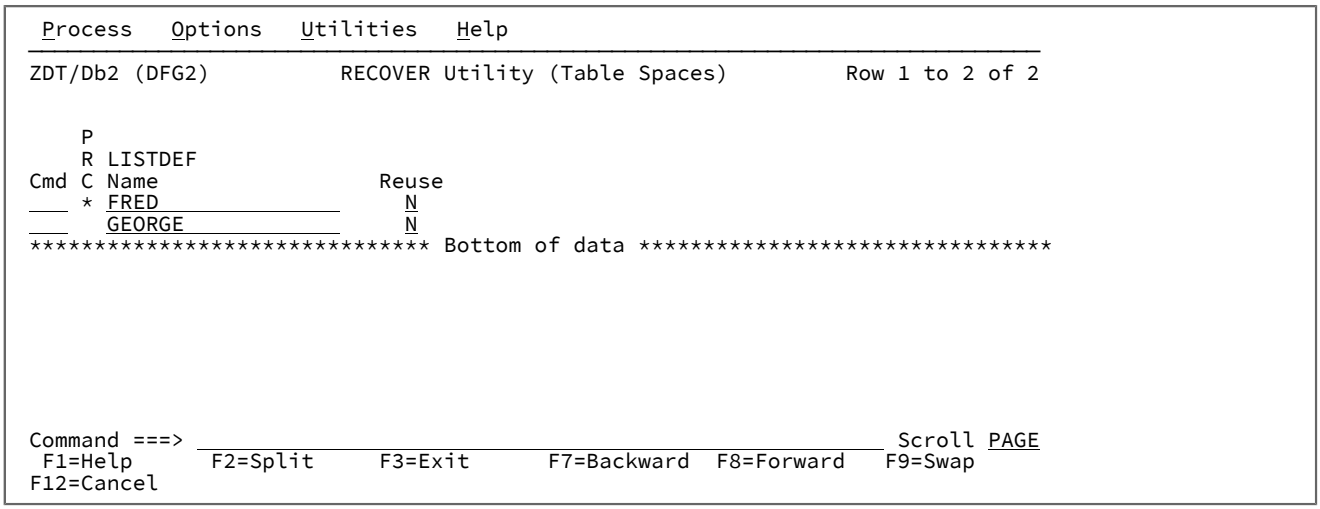
Related tasks

- RECOVER (table spaces) on page 368

RECOVER Utility (Table Spaces) with LISTDEF panel

You use the RECOVER Utility (Table Spaces) with LISTDEF panel to generate utility control statements using the RECOVER INDEXSPACE form of the RECOVER utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- RECOVER utility on page 368
- RECOVER (table spaces) on page 368

Recovery Information panel

ZDT/Db2 displays the Recovery Information panel if you use the ICS line command against an item listed on:

- The Databases panel on page 609, or
- The Table Spaces panel on page 879, or
- The Indexes panel on page 723

Related tables panel

You use the Related tables panel to display information about the tables related to the table being edited when you issued the REDIT command.

The panel displays a selection list showing all the Db2® objects that are directly related to the object being edited by an RI constraint. This includes tables that are parent tables in a relationship with the table being edited, and tables that are dependent tables in a relationship with the table being edited.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFE2)        Related tables for: DSN8910.EMP                Top of 4
4 rows fetched                                                      Format TABL
       RELATED  RELATED                                    COUNT
       TABLE    TABLE      RELATIONSHIP RELATIONSHIP  OF FOREIGN DELETE
SEL    OWNER    NAME       NAME         TYPE         KEY COLUMNS RULE
       *        *          *            *                      * *
----   #1--+--- #2--+----- #3--+----1-- #9--+----         ----#6 #10-+---
****  Top of data  ****
_____   DSN8910  DEPT       RDE          DEPENDENT             1 SET NULL
_____   DSN8910  EMPPROJACT REPAE        DEPENDENT             1 RESTRICT
_____   DSN8910  PROJ       RESPEMP      DEPENDENT             1 RESTRICT
_____   DSN8910  DEPT       RED          PARENT                1 SET NULL
****  End of data  ****




Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

The entries shown on the Related tables panel depend on the cursor position when you issued the REDIT command:

- If the cursor was not on the data for a column that is part of a foreign, parent or primary key, all related tables are shown.
- If the cursor was on the data for a column that is part of a foreign, parent or primary key, only the related tables that include that column in a foreign, parent or primary key are shown.

The columns that are displayed include the SEL field, and columns of SYSIBM.SYSRELS.

SEL

Selection column. You can issue any of these line commands against the Db2® objects in the list:

A

Alter table.

ALS

Show aliases.

B

Browse table. (See Note 1)

C

Copy table.

CDI

Show column distribution statistics.

CH

Show child (dependent) tables.

CHR

Show child (dependent) relations.

CK

Show table check constraints.

COL

Show columns.

CON

Show table (unique and primary) constraints.

D

Show database.

E

Edit table. (See Note 1)

EE

Edit table - new session. (See Note 3)

FK

Show foreign keys.

I

Details about table or view.

KEY

Show relationship keys. (See Note 2)

P

Show privileges.

PA

Show parent tables.

PAR

Show parent relations.

PK

Show primary key.

PKG

Show packages.

PL

Show plans.

PR

Print table.

PST

Show partition statistics.

ROW

Show all columns for this row.

S

Show table spaces.

SYN

Show synonyms.

V

View table. (See Note 1)

VE

View table - new session. (See Note 3)

VW

Show views.

X

Show indexes.

XC

Show indexes and index columns.

Note:

1. The E, B, and V commands edit, browse, or view the object by stacking a new ZDT/Db2 editor session on top of the existing ZDT/Db2 editor session. You cannot return to the original ZDT/Db2 editor session without first ending the new ZDT/Db2 editor session.
2. For a parent relationship type, the KEY command shows the primary keys for the two tables in the relationship.

   For a dependent relationship type, the KEY command shows the foreign key columns for the two tables in the relationship.
3. The EE and VE commands edit (or view) the object by starting a new ISPF logical session (split screen) and then starting a Z Data Tools/Db2 editor session in the new logical session.
   - When the REDIT command was issued and the cursor was not located on the data for a column that is part of a parent, primary or foreign key, the new ZDT/Db2 editor session shows all rows of the table.
   - When the REDIT command was issued with the cursor located on the data for a column that is part of a parent, primary or foreign key, the new ZDT/Db2 editor session only shows those rows related to the row where the cursor was located.

   The ZDT/Db2 editor session of the related table may have no rows when:
   - For a dependent table, there are no rows in the table with a matching foreign key value.
   - For a parent table, the foreign key value contains one or more columns with the Db2® null value.

Parent panels

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Child panels

Various. Depends on which line command entered.

Related tasks

- Specifying a Db2 object name on page 46
- Listing related tables on page 193
- Displaying or printing a list of objects on page 304
- Db2 Location Selection panel on page 621

Remote Db2® Location Selection panel

You use the Remote Db2® Location Selection panel to display all the available remote locations. You can only access remote locations that have been defined to your local Db2® subsystem.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)          Remote Db2 Location Selection          Row 1 to 4 of 4

Sel Location
    *_______________

_   QXPMVS8DB2D
_   QXPMVS8DB26
_   QXPMVS6DB2D
_   QXPMVS8DB62
****************************** END OF Db2 DATA ********************************




Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F12=Cancel
```

Sel

Selection column. Type `s` to select a location.

Location

The name of the location.

Parent panels

An asterisk (*) or pattern entered in the **Location** field on any Object List utility panel.

Child panels

None.

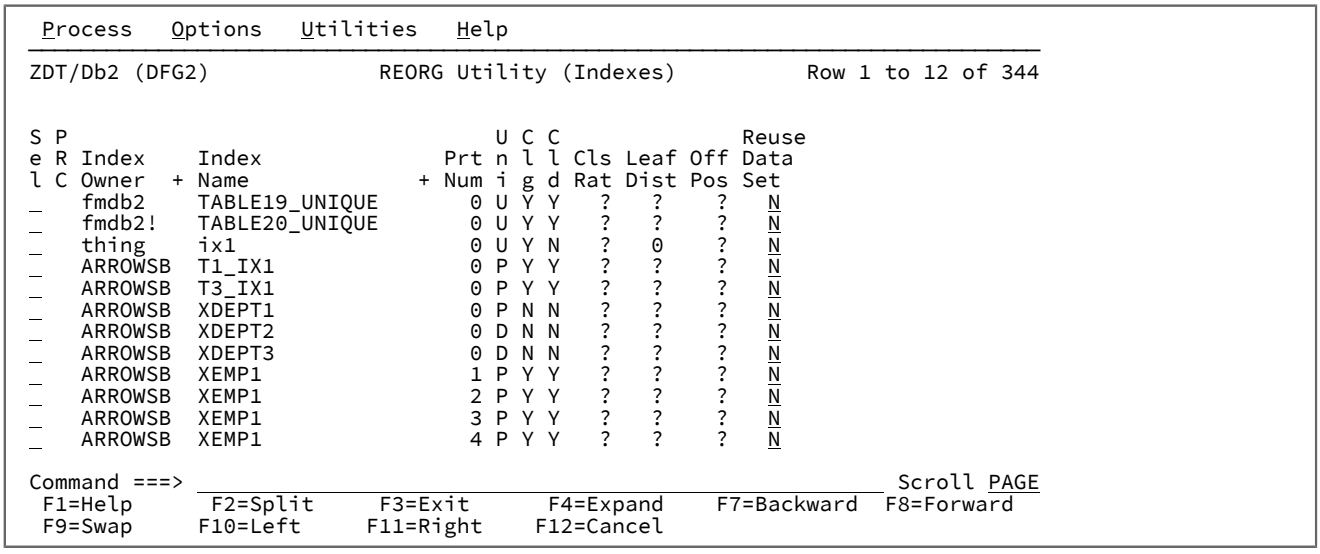
Related tasks

- Specifying a Db2 object name on page 46
- Displaying or printing a list of objects on page 304
- Db2 Location Selection panel on page 621

REORG Utility (Indexes) panel

You use the REORG Utility (Indexes) panel to generate utility control statements using the REORG INDEX form of the REORG utility.

Panel and field definitions



```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)             REORG Utility (Indexes)          Row 1 to 12 of 344


S P                                  U C C              Reuse
e R Index    Index              Prt n l l Cls Leaf Off Data
l C Owner  + Name             + Num i g d Rat Dist Pos Set
_   fmdb2    TABLE19_UNIQUE       0 U Y Y   ?    ?    ?   N
_   fmdb2!   TABLE20_UNIQUE       0 U Y Y   ?    ?    ?   N
_   thing    ix1                  0 U Y N   ?    0    ?   N
_   ARROWSB  T1_IX1               0 P Y Y   ?    ?    ?   N
_   ARROWSB  T3_IX1               0 P Y Y   ?    ?    ?   N
_   ARROWSB  XDEPT1               0 P N N   ?    ?    ?   N
_   ARROWSB  XDEPT2               0 D N N   ?    ?    ?   N
_   ARROWSB  XDEPT3               0 D N N   ?    ?    ?   N
_   ARROWSB  XEMP1                1 P Y Y   ?    ?    ?   N
_   ARROWSB  XEMP1                2 P Y Y   ?    ?    ?   N
_   ARROWSB  XEMP1                3 P Y Y   ?    ?    ?   N
_   ARROWSB  XEMP1                4 P Y Y   ?    ?    ?   N

Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Sel

Selection column. Type s to select an index.

PRC

Processing column. An asterisk (*) indicates the selected index.

Index Creator

The name of the creator of the index.

Index Name

The name of the index.

Part Num

The partition number for the index. For a non-partitioned index, this is 0. Partitioned indexes appear as multiple rows, one row for each partition of the index. This is a display-only field

Uni

The UNIQUERULE flag for the index, from the SYSINDEXES catalog table

Clg

The CLUSTERING flag for the index, from the SYSINDEXES catalog table.

Cld

The CLUSTERED flag for the index, from the SYSINDEXES catalog table.

Cls Rat

The CLUSTERRATIO flag for the index, from the SYSINDEXES catalog table.

Leaf Dist

The LEAFDIST flag for the index, from the SYSINDEXPART catalog table.

Off Pos

This is a display-only field. This value indicates the number of rows relocated out of their optimal position for the index. If the CARD value for the index is greater than zero, the value is calculated using the expression:

(AVG(NEAROFFPOS + FAROFFPOS) * 100)/CARD

This value helps you identify indexes that might require organization. For more information, see the *DB2 Utility Guide and Reference*.

Reuse Data set

Whether Db2® physically deletes and redefines the data sets for the index:

Y

Data set for the index reused.

N

Data set for the index not reused. This is the default setting.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

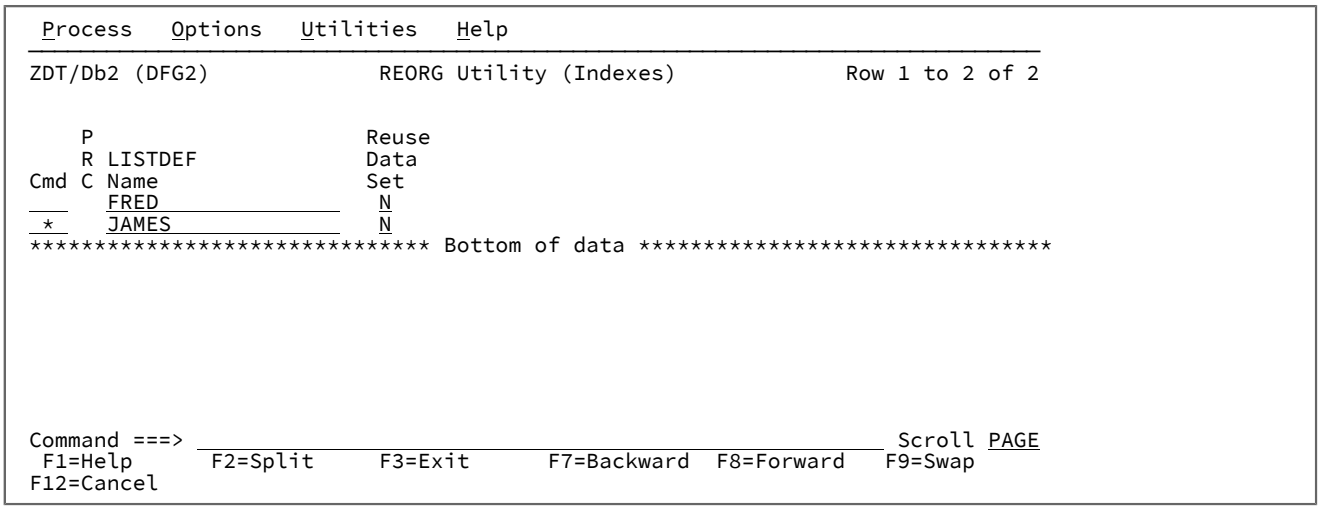
Related tasks

- REORG (indexes) on page 374

REORG Utility (Indexes) with LISTDEF panel

You use the REORG Utility (Indexes) with LISTDEF panel to generate utility control statements using the REORG INDEX form of the REORG utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- REORG utility on page 371
- REORG (indexes) on page 374

REORG Utility (Table Spaces) panel

You use the REORG Utility (Table Spaces) panel to generate utility control statements using the REORG TABLESPACE form of the REORG utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)           REORG Utility (Table Spaces)       Row 1 to 12 of 229


S P          Table        Avg Avg Unload        Reuse
e R Database Space    Prt Ind Off External Log  Data
l C Name     Name     Num Ref Pos (Y/N)   (Y/N) Set
_   BILLSDB1 TS1D       0   ?   ?   N       N     N
_   BILLSDB1 TS1E       1   ?   ?   N       N     N
_   BILLSDB1 TS1E       2   ?   ?   N       N     N
_   BILLSDB1 TS1E       3   ?   ?   N       N     N
_   BILLSDB1 TS1E       4   ?   ?   N       N     N
_   BILLSDB2 UTILTS1    0   ?   ?   N       N     N
_   CJMAODA  CJMAOSD    0   0   0   N       N     N
_   CJMAODA  CJMAOSE    1   0   0   N       N     N
_   CJMAODA  CJMAOSE    2   ?   ?   N       N     N
_   CJMAODA  CJMAOSE    3   0   0   N       N     N
_   CJMAODA  CJMAOSE    4   ?   ?   N       N     N
_   CJMAODA  CJMAOSP    0   0   0   N       N     N

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Sel

Selection column. Type s to select a table space.

PRC

Processing column. An asterisk (*) indicates the selected table space.

Database Name

The name of the database in which the table space is defined.

Table Space Name

The name of the table space.

Prt Num

The partition number for the table space. This is 0 for a non-partitioned table space. Partitioned table spaces appear as multiple rows, one row for each partition of the table space. This is a display-only field.

Avg Ind Ref

This is a display-only field. This is an average value for the number of rows relocated out of their original position. If the table space/partition's cardinality is greater than zero, the calculated value is:

(AVG(NEARINDREF + FARINDREF) * 100)/CARD

This value helps you identify table spaces that might require organization. For more information about these values, see the *DB2 Utility Guide and Reference*.

Avg Off Pos

This is a display-only field. This is an average value for the number of rows relocated out of their optimal position for all indexes within the table space or partition. The average is for all indexes which have a CARD value greater than zero, using the expression:

(AVG(NEAROFFPOS + FAROFFPOS) * 100)/CARD

This value helps you identify table spaces that might require organization. For more information, see the *DB2 Utility Guide and Reference*.

Unload External (Y/N)

Whether the REORG process leaves the data in external format and therefore acts as an *unload utility*:

Y

REORG processing halts at the unload phase and leaves the data in external format in the SYSREC00 data set. Load control cards are produced in the SYSPUNCH data set. The data produced is in a format suitable for loading into any similar table structure, in any Db2® subsystem. The DD names in the job and resultant SYSPUNCH are compatible with the DSNTIAUL sysrec naming (for example, SYSREC00), therefore the ZDT/Db2 LOAD Utility can accept both formats, without requiring any JCL editing before submission.

N

Full REORG performed.

Log (Y/N)

LOG option:

Y

Changes written to the Db2® log.

N

Changes not written to the Db2® log. This is the default setting.

For more information, see the *DB2 Utility Guide and Reference*.

Reuse Data Set

Reuse option:

Y

The data set for the table space is not deleted and re-defined as part of the RECOVER utility processing.

N

The data set for the table space is deleted and redefined as part of the RECOVER utility processing.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- REORG (table spaces) on page 372

REORG Utility (Table Spaces) with LISTDEF panel

You use the REORG Utility (Table Spaces) with LISTDEF panel to generate utility control statements using the REORG TABLESPACE form of the REORG utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)           REORG Utility (Table Spaces)          Row 1 to 2 of 2


    P                     Unload        Reuse
    R LISTDEF             External Log  Data
Cmd C Name                (Y/N)   (Y/N) Set
___ * FRED                  N       N    N
___   JAMES                 N       N    N
****************************** Bottom of data ********************************









Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- REORG utility on page 371
- REORG (table spaces) on page 372

Revoke privileges panels

These panels are:

- Revoke Application Plan Privileges panel
- Revoke Buffer Pool Privileges panel
- Revoke Collection Privileges panel
- Revoke Column Privileges panel
- Revoke Database Privileges panel
- Revoke Distinct Type Privileges panel
- Revoke Function Privileges panel

- Revoke Package Privileges panel
- Revoke Schema Privileges panel
- Revoke Sequence Privileges panel
- Revoke Storage Group Privileges panel
- Revoke Stored Procedure Privileges panel
- Revoke System Privileges panel
- Revoke Table Privileges panel
- Revoke Table Space Use Privileges panel

ZDT/Db2 displays one of these panels if you use the Privileges utility function (3.5) to revoke privileges for an object type, or if you specify the R line command against an object displayed in a list of Db2® objects.

Format of revoke privilege panels

The revoke privileges panels are similar to the corresponding grant privilege panels. However, in place of the **To** field, the revoke privileges panels contain the following fields:

- A **From** field. Use this field to specify the SQLID of the user for whom the privileges are to be revoked (the revokee).
- A **By** field. Use this field to specify the SQLID of the user who granted the privileges (the grantor), or *"ALL"* to indicate all grantors.

Related tasks

Routine Parameters panel

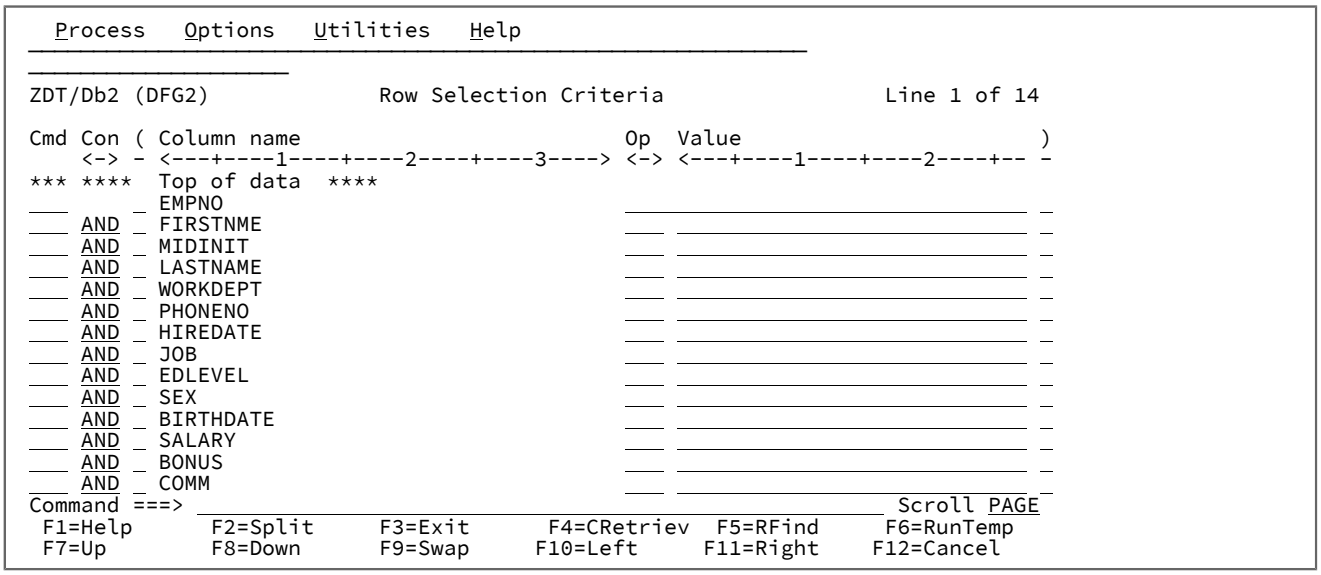
ZDT/Db2 displays the Routine Parameters panel if you use the PARM line command against an item listed on:

- The Functions panel on page 709, or
- The Stored Procedures panel on page 852

Row Selection Criteria panel

You use the Row Selection Criteria panel to specify row selection criteria by column.

Panel and field definitions



Cmd

Prefix command column.

You can use the following prefix commands when editing row selection criteria by column:

A

Identifies the line after which data is to be moved or copied.

B

Identifies the line before which data is to be moved or copied.

C

Copy one line.

C*n*

Copy *n* rows.

CC

Copy block of lines. Mark start and end of block.

D

Delete one line.

D*n*

Delete *n* lines.

DD

Delete block of lines. Mark start and end of block.

M

Move one line.

M*n*

Move *n* lines.

MM

Move block of lines. Mark start and end of the block.

R

Repeat one line.

R*n*

Repeat *n* lines.

RR

Repeat block of lines. Mark start and end of the block.

RR*n*

Repeat block of lines *n* times. Mark start and end of block.

Con

Connector column.

You specify the AND and OR connectors in this column. The default value is AND. You can use abbreviations. For example, if you overtype the existing connector, AND, with `o`, ZDT/Db2 generates an OR connector.

Connectors are only relevant to lines containing expressions (that is, where you have specified an operator in the **Op** column).

You can overtype the existing connector with one of the following values:

AND or &

ZDT/Db2 generates an AND connector in the resultant WHERE clause.

OR or |

ZDT/Db2 generates an OR connector in the resultant WHERE clause.

(

Left parenthesis column.

If you need to define the sequence of evaluation for the resultant WHERE clause by enclosing part of the clause in parentheses, use this column to type a left parenthesis ("(") at the appropriate point.

You can only use a parenthesis on a line containing expressions (that is, where you have specified an operator in the **Op** column). The number of left parentheses must match the number of right parentheses.

Column name

The name of the column.

On 80-byte screens, ZDT/Db2 displays either the column name details (the default) or the data type details for the column. To toggle between ZDT/Db2 displaying the **Column name** and **Data type(length)** details, press the PF11 function key.

Data type(length)

The Db2® data type and, where applicable, length of the column.

On 80-byte screens, ZDT/Db2 displays either the column name details (the default) or the data type details for the column. To toggle between ZDT/Db2 displaying the **Column name** and **Data type(length)** details, press the PF11 function key.

Op

Operator column.

Operators are used to create expressions. When you type an operator in the **Op** column, it indicates that you want ZDT/Db2 to form an expression using the column name on that line of the panel, the operator you have specified and, with the exception of the operators NL, ¬NL, and NNL, the contents of the **Value** column.

ZDT/Db2 supports the following operators:

= or EQ

True if the terms are equal (numerically, or when padded).

¬= or NE

True if the terms are not equal (inverse of =)

> or GT

Greater than

< or LT

Less than

>= or GE

Greater than or equal to

\= or ¬<

Not less than

<= or LE

Less than or equal

\> or ¬>

Not greater than

LK

LIKE predicate. Column must not be numeric.

¬LK or NLK

NOT LIKE. Column must not be numeric.

BTW

BETWEEN. **Value** must be in the form *value1*,*value2*.

¬BT or NBT

NOT BETWEEN. **Value** must be in the form *value1*,*value2*.

IN

IN. **Value** must be in the form *value1*,*value2*,...

¬IN or NIN

NOT IN. **Value** must be in the form *value1*,*value2*,...

NL

IS NULL. **Value** must be blank.

¬NL or NNL

IS NOT NULL. **Value** must be blank.

RG

Range. **Value** must be in the form *value1*,*value2*.

NR

Not in range. **Value** must be in the form *value1*,*value2*.

Value

The character or numeric constant to be used with the specified operator and column name to create an expression in the resultant WHERE clause.

When the operator is NL, ¬NL, or NNL, the Value column must be blank. For all other operators, the Value column must not be blank

For operators that support more than one value (BTW, ¬BT, NBT, IN, ¬IN, NIN, RG, or NR), the values must be separated by commas. For operators IN, ¬IN, or NIN, and the column is alphanumeric, you must enclose each value in quotation marks.

If the data type of the column is alphanumeric, the value can be:

- A character string not starting or ending with a single quotation mark. and not containing any embedded quotation marks or commas. The case of the string is ignored.
- A character string enclosed in single quotation marks. The string can contain blanks and commas, but not any embedded quotation marks. The case of the string is ignored.

If the data type of the column is numeric, the value must be numeric.

For DATE, TIME, and TIMESTAMP columns, if you specify a "=" in the **Value** field, ZDT/Db2 displays the current date, time, or timestamp values respectively in the required format.

)

Right parenthesis column.

If you need to define the sequence of evaluation for the resultant WHERE clause by enclosing part of the clause in parentheses, use this column to type a right parenthesis (")") at the appropriate point.

You can only use parentheses on lines containing expressions (that is, where you have specified an operator in the **Op** column). The number of left parentheses must match the number of right parentheses.

Parent panels

- Column Selection/Edit panel on page 497

Child panels

None.

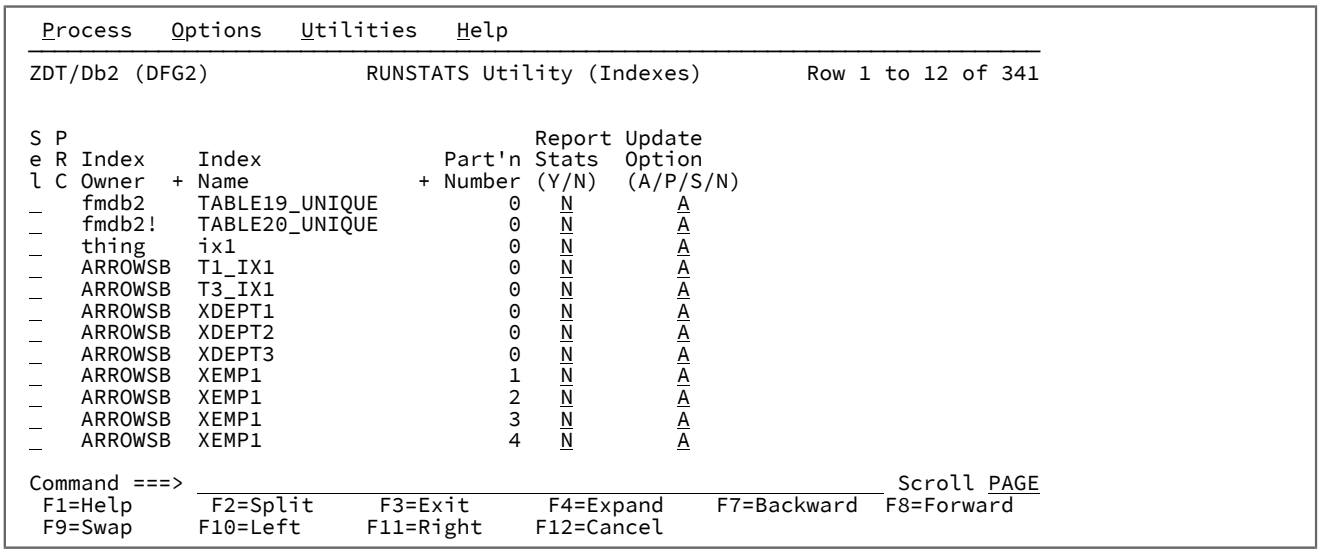
Related tasks

- Selecting rows on page 86

RUNSTATS Utility (Indexes) panel

You use the RUNSTATS Utility (Indexes) panel to generate utility control statements using the RUNSTATS INDEX form of the RUNSTATS utility.

Panel and field definitions



Sel

Selection column. Type `s` to select an index.

PRC

Processing column. An asterisk (*) indicates the selected index.

Index Creator

The name of the creator of the index.

Index Name

The name of the index.

Part'n Number

The partition number for the index. A zero indicates a non-partitioned index. A non-zero value indicates the partition number of a partitioned index.

Report Stats (Y/N)

Whether a report outlining the results of the utility is written to the SYSPRINT DDNAME:

Y

Report written to the SYSPRINT DDNAME.

N

Report not written to the SYSPRINT DDNAME. This is the default setting.

For more information, see the *DB2 Utility Guide and Reference*.

Update Option (A/P/S/N)

Indicates which catalog tables (if any) are updated by the RUNSTATS utility. The available options are:

A

All. All catalog tables are updated. This is the default setting.

P

Access path only. Statistics related to access path selection are updated

S

Space only. Only statistics related to space information are updated

N

None. No catalog tables are updated.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- RUNSTATS (indexes) on page 378

RUNSTATS Utility (Indexes) with LISTDEF panel

You use the RUNSTATS Utility (Indexes) with LISTDEF panel to generate utility control statements using the RUNSTAT INDEX form of the RUNSTAT utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)            RUNSTATS Utility (Indexes)          Row 1 to 2 of 2


    P                    Report Update
    R LISTDEF            Stats  Option
Cmd C Name               (Y/N)  (A/P/S/N)
___ * FRED                 N        A
___   JAMES                N        A
****************************** Bottom of data ********************************









Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

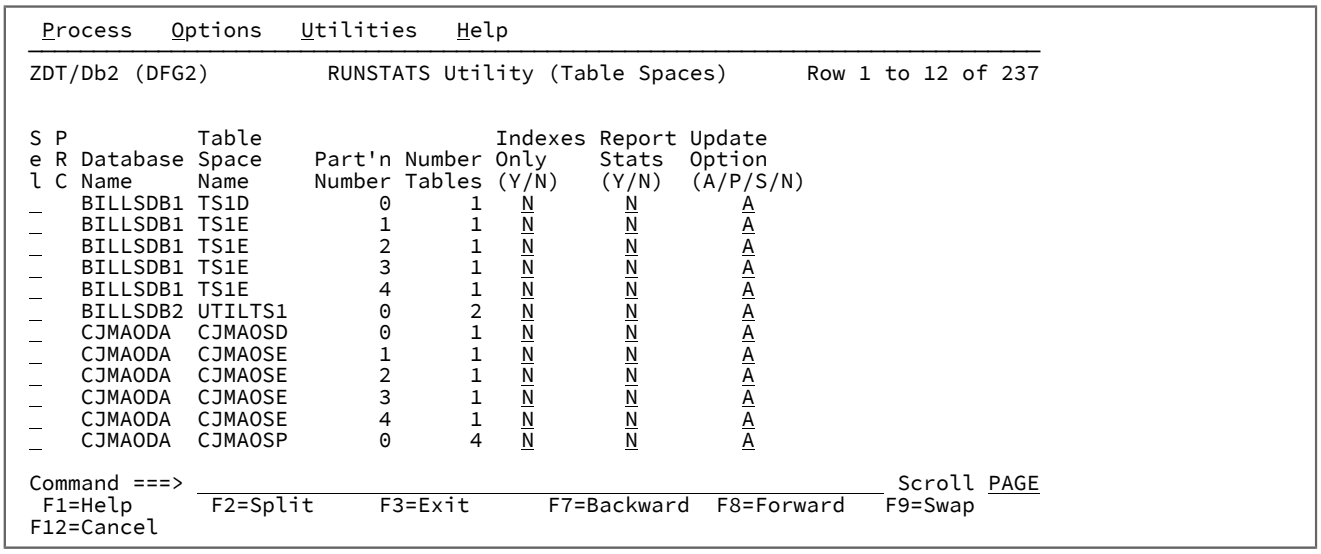
Related tasks

- RUNSTATS utility on page 376
- RUNSTATS (indexes) on page 378

RUNSTATS Utility (Table Spaces) panel

You use the RUNSTATS Utility (Table Spaces) panel to generate either RUNSTATS TABLESPACE or RUNSTATS INDEX control statements, depending on the value selected for the **Indexes Only** column.

Panel and field definitions



Sel

Selection column. Type `s` to select a table space.

PRC

Processing column. An asterisk (*) indicates the selected table space.

Database Name

The name of the database in which the table space is defined.

Table Space Name

The name of the table space.

Part'n Number

The partition number for the table space. A zero value indicates a non-partitioned table space; a non-zero value indicates the partition number.

Number Tables

The number of tables in the table space.

Indexes Only (Y/N)

Whether RUNSTATS is performed for only the indexes in the table spaces (without gathering statistics for the tables in the table space itself):

Y

RUNSTATS performed for indexes only.

N

RUNSTATS performed for tables and indexes. This is the default setting.

Report Stats (Y/N)

Whether a report outlining the results of the utility is written to the SYSPRINT DDNAME:

Y

Report written to the SYSPRINT DDNAME.

N

Report not written to the SYSPRINT DDNAME. This is the default setting.

For more information, see the *DB2 Utility Guide and Reference*.

Update Option (A/P/S/N)

Indicates which catalog tables (if any) are updated by the RUNSTATS utility:

A

All. All catalog tables are updated. This is the default setting.

P

Access path only. Statistics related to access path selection are updated.

S

Space only. Only statistics related to space information are updated.

N

None. No catalog tables are updated.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

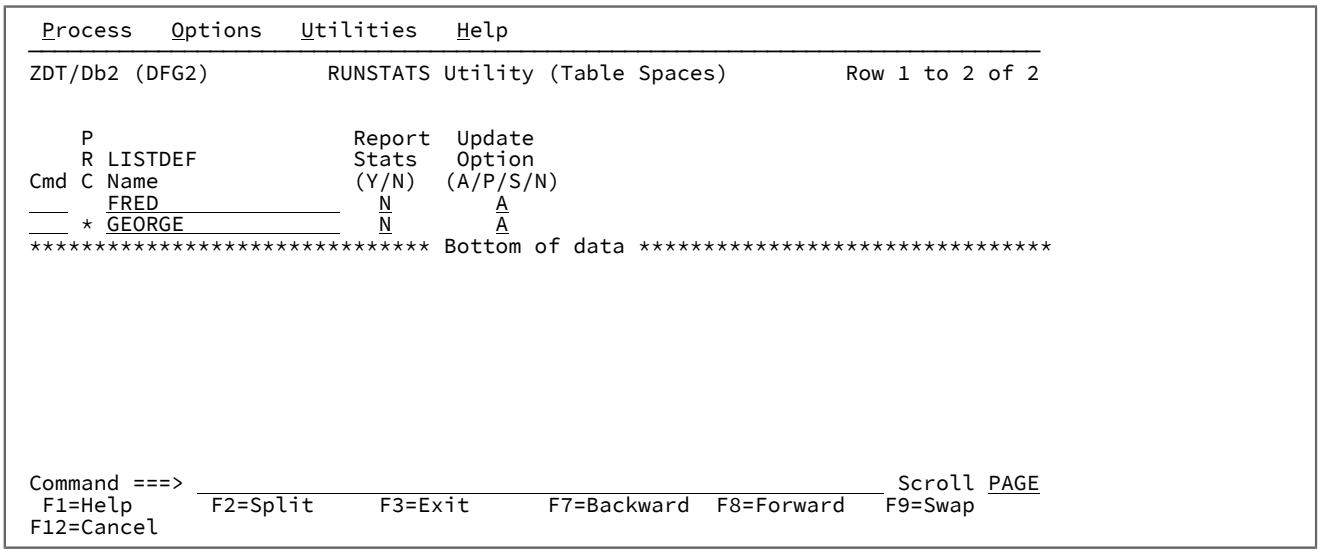
Related tasks

- RUNSTATS (table spaces) on page 377

RUNSTATS Utility (Table Spaces) with LISTDEF panel

You use the RUNSTATS Utility (Table Spaces) with LISTDEF panel to generate utility control statements using the RUNSTAT TABLESPACE form of the RUNSTAT utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- RUNSTATS utility on page 376
- RUNSTATS (table spaces) on page 377

Schemas panel

You use the Schemas panel to list information about the schemas you have selected.

Panel and field definitions

The columns that are displayed include the SEL field and columns of various catalog tables.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                      Schemas
Location:                                                    Format TABL


SEL    SCHEMA     UDTS   UDFS   STPS   TRGS
       *_______ _____* _____* _____* _____*
----   #2--+--- ----#3 ----#4 ----#5 ----#6
****  Top of data  ****
_____  SHRIKE2       1      3      0      2
_____  SHRIKES       1      4      0      1
_____  SQLJ          0      0      7      0
_____  SYSPROC       0      0     10      0
_____  SYSIBM        0      0     13      0
****  End of data  ****




Command ===> _____________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Distinct Types panel on page 637
- Functions panel on page 709
- Stored Procedures panel on page 852
- Triggers panel on page 903

Child panels

To display this panel...	Use/do this
Distinct Types panel on page 637	Line command DT
Functions panel on page 709	Line command F
Generate SQL From Db2 Catalog panel on page 712	Line command GEN
Stored Procedures panel on page 852	Line command O
Privileges panels on page 768	Line command P
Revoke privileges panels on page 798	Line command R
Display Row panel on page 637	Line command ROW
Triggers panel on page 903	Line command TR

To display this panel...	Use/do this
	Primary command SORT

Sort Fields panel on page 841

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Scramble Exit Specification panel

The Scramble Exit Specification panel allows you to to provide the name of a scramble exit, a constant to be passed to the exit, and to specify whether numeric fields are to be formatted before the value is passed to the exit.

Panel and field definitions

Figure 130. Scramble Exit Specification panel

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)           Scramble Exit Specification

Fixed Attributes:
         Column name  COLCARD
                Type  INTEGER

Scramble Exit:
        Program name  _________
   Optional constant  ____________________________________________  +

Scramble Exit Processing Options:
 Format Option
 1 1. None
   2. Format
   3. Leading Zeros



Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Program name

A 1 to 8 character valid load module member name that identifies the scramble exit invoked during a copy operation for this output column. The exit must be in the form of a load module, in any load library available to ZDT/Db2 at the time of the copy process, either by a STEPLIB DD statement, or in LINKLIST, or LPALIST.

Optional constant

A constant value that is passed to the exit for each call type. To enter a hexadecimal value, press F4 or enter the EXPAND command with the cursor positioned in this field and use the command HEX ON from the expand window.

Format Option

This option is only displayed for numeric columns and allows you to request ZDT/Db2 to pass the value specified in **Optional constant** to the exit in one of the following forms:

1. **None**

   Field value passed to the exit in the internal format.

2. **Format**

   Field value passed to the exit as displayed numeric with leading zeros suppressed.

3. **Leading Zeros**

   Field value passed to the exit as displayed numeric with leading zeros.

Parent panels

- Column Attributes panel (alphanumeric) on page 484
- Column Attributes panel (numeric) on page 490

Related tasks

- Scrambling data on page 292

Select Statement Browse panel

ZDT/Db2 displays the Select Statement Browse panel if the **Use edit (instead of browse) to display results** option is not selected, and you press Enter when one of these panels is displayed:

- Basic SELECT Prototyping panel on page 476
- Advanced SELECT Prototyping panel on page 451

Related tasks

- Systems Options (1 of 4) panel on page 857

Select Statement Edit panel

ZDT/Db2 displays the Select Statement Edit panel if the **Use edit (instead of browse) to display results** option is selected, and you press Enter when one of these panels is displayed:

- Basic SELECT Prototyping panel on page 476
- Advanced SELECT Prototyping panel on page 451

Related tasks

- Systems Options (1 of 4) panel on page 857

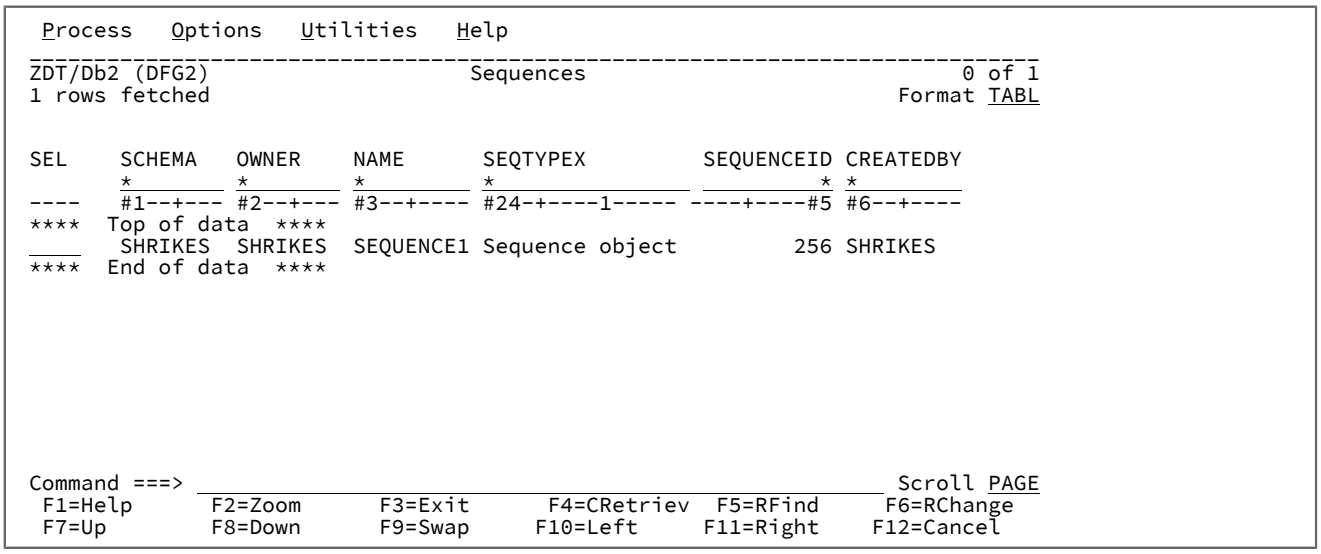
Sequences panel

You use the Sequences panel to display information about the sequences you have selected.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSSEQUENCES.

Figure 131. Sequences panel



To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748

Child panels

To display this panel…	Use/do this
	Line command CR
Create Sequence panel	
	Line command DEP
Sequence Dependencies panel	
	Line command DR
Drop Sequence panel	
	Line command G
Grant Sequence Privileges panel	
	Line command ROW
Display Row panel on page 637	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Related tasks

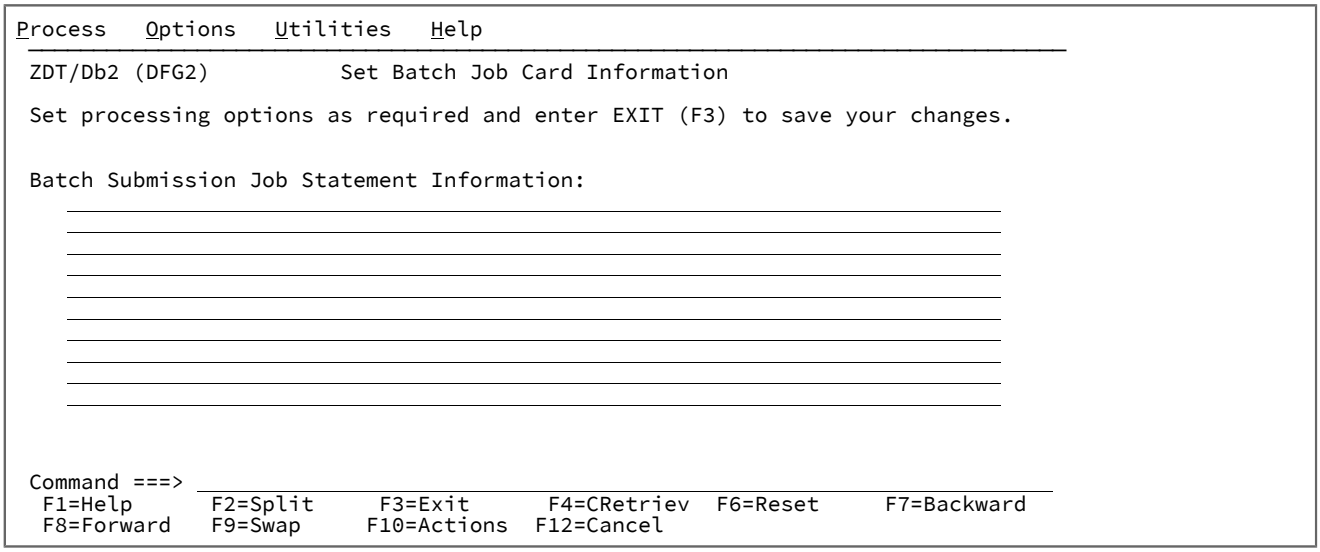
- Managing tables used by SQL explain on page 430

Set Batch Job Card Information panel

You use the Set Batch Job Card Information panel to specify a JCL jobcard that ZDT/Db2 is to use whenever it generates a batch job. You can also specify a JOBLIB or other statements.

These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



Batch Submission Job Statement Information

The JCL job card information that is used for any batch job that is created.

Child panels

None.

Related tasks

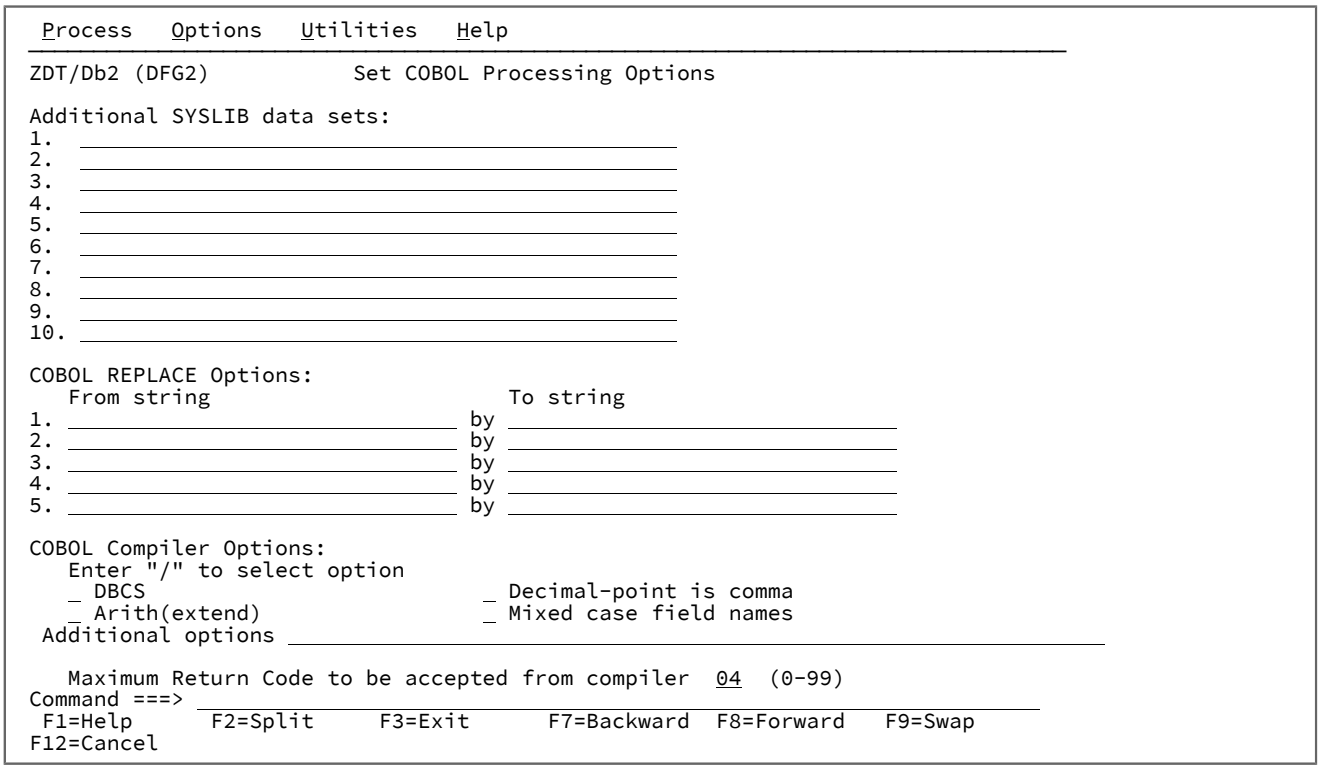
- Job card specifications (option 0.3) on page 65

Set COBOL Processing Options panel

You use the Set COBOL Processing Options panel to customize how ZDT/Db2 compiles COBOL copybooks to generate ZDT/Db2 templates.

These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



Additional SYSLIB data sets

When you use one of the preceding ZDT/Db2 functions, if the COBOL copybook that you specify uses COPY compiler-directing statements to include other members that do not belong to the same PDSs as the copybooks, then you need to specify those other PDSs here.

Here's why: members included by COPY compiler-directing statements must be found in the SYSLIB concatenation. When you specify a copybook on a Z Data Tools/Db2 panel, the SYSLIB concatenation consists of the PDS of the copybook, plus up to ten additional PDSs that you can specify on this panel. These PDSs are searched in order (PDS of the copybook, followed by these additional PDSs, 1–10.)

COBOL REPLACE options

"From" and *"To"* pseudo-text character strings for the REPLACE compiler-directing statement.

If the COBOL copybook contains characters that you want to replace with other characters before compiling the copybook into a template, then use these replacing options.

For example, if the copybook contains colon characters (:) that you want to remove before compiling, then specify `==:==` as a *"From"* string and `=====` as the matching *"To"* string.

For details on specifying *"From"* and *"To"* strings for REPLACE, see the *IBM COBOL Language Reference*.

COBOL Compiler Options

Compiler options to be used when compiling a COBOL copybook into a template:

DBCS

Instructs Z Data Tools to use the DBCS compiler option.

Decimal-point is comma

Uses the "Decimal-point is comma" SPECIAL-NAMES paragraph when compiling COBOL copybooks.

Arith(extend)

Uses the Arith(extend) COBOL compile option when compiling COBOL copybooks.

Mixed case field names

Retains the original case of the field name as coded in the COBOL copybook.

Maximum Return Code

Sets the maximum warning/error code accepted from the compiler, when compiling a copybook. Any code higher than specified causes ZDT/Db2 to display a pop-up panel, in which you can review the compilation and decide about future processing.

Additional options

Additional COBOL compiler options which are added via the CBL statement when a COBOL compile is run to build or update a template. These options are validated during the compilation process. To avoid compilation errors, ensure that the syntax is correct and that any additional data sets required by these options are allocated before invoking Z Data Tools.

Child panels

None.

Related tasks

- COBOL compiler specifications (option 0.5) on page 65

Set HLASM Processing Options panel

The HLASM Processing Options panel controls various options used by the HLASM compiler to generate templates from your HLASM copybooks.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
 Z Data Tools             Set HLASM Processing Options

 Additional SYSLIB data sets:
 1.  _____________________________________________
 2.  _____________________________________________
 3.  _____________________________________________
 4.  _____________________________________________
 5.  _____________________________________________
 6.  _____________________________________________
 7.  _____________________________________________
 8.  _____________________________________________
 9.  _____________________________________________
 10. _____________________________________________

HLASM Compiler Options:

   Enter "/" to select option
   _  DBCS
   _  NOALIGN

   Additional options ____________________________________________________

   Maximum Return Code to be accepted from compiler  04  (0-99)
Command ===>_____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward  F9=Swap
 F10=Actions  F12=Cancel
```

Additional SYSLIB data sets

A concatenation of up to 10 SYSLIB data sets may be entered, these data sets are searched in the order specified for COPY or INCLUDE members for the compilation.

HLASM Compiler Options for Copybooks

Compiler options to be used when compiling a HLASM copybook into a template:

DBCS

If selected, use the DBCS compiler option.

NOALIGN

If selected, use the NOALIGN compiler option.

Additional options

Additional HLASM compiler options which are added via the *PROCESS statement when the assembler is run to build or update a template. These options are validated during the assembly process. To avoid assembly errors please ensure the syntax is correct.

Parent panels

- Set System Processing Options panel on page 835
- Set Processing Options panel on page 832

Child panels

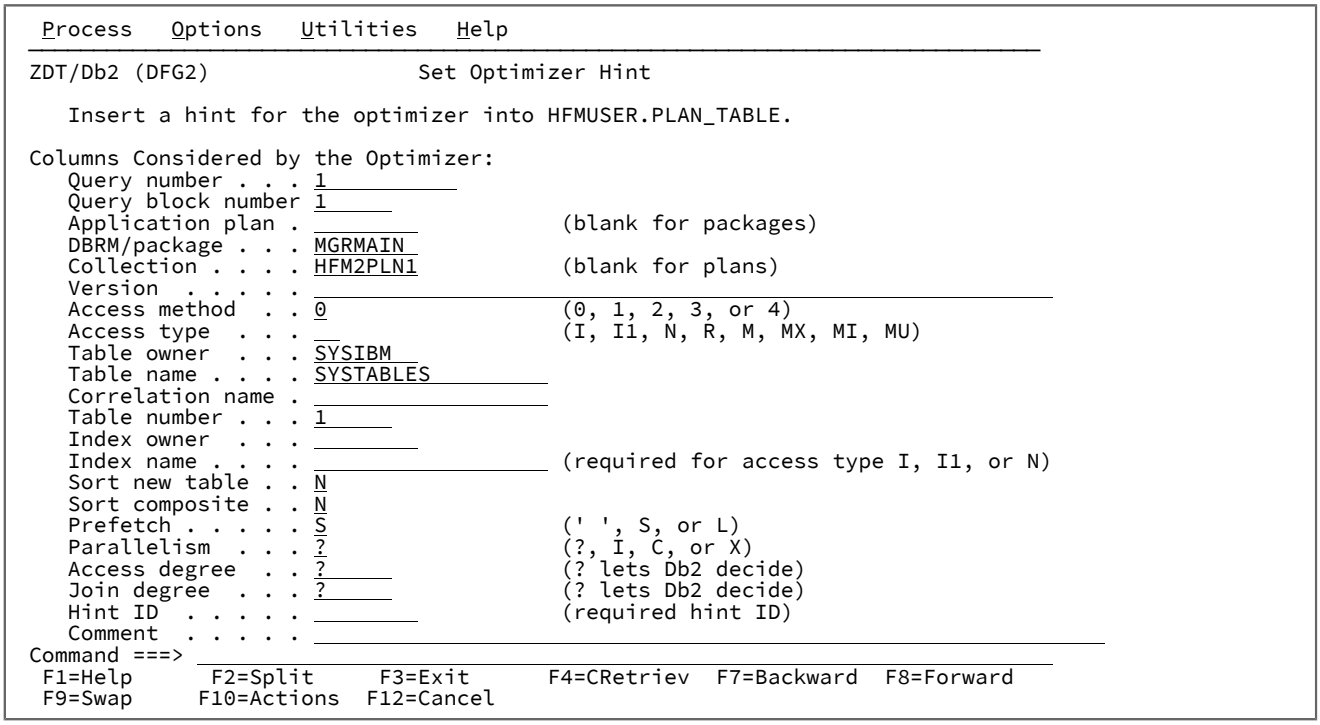
- None.

Comments

- HLASM compiler specifications (option 0.6) on page 66

Set Optimizer Hint panel

Panel and field definitions



Parent panels

- Plan Table Rows panel on page 754

Child panels

None.

Related tasks

Set Output Data Set Allocation Options panel

Panel and field definitions

Figure 132. Set Output Data Set Allocation Options panel

```
  Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
Z Data Tools        Set Output Data Set Allocation Options

Set processing options as desired and enter EXIT (F3) to save your changes.
Enter RESET to restore installation defaults.

Data set  Space  Primary   Secondary Management Storage   Data      Device
          Units  Quantity  Quantity   Class     Class     Class     Type

Audit Log  CYL   16        8         _________ _________ _________ SYSALLDA

Print      CYL   64        32        _________ _________ _________ SYSALLDA

Trace      CYL   64        32        _________ _________ _________ SYSALLDA

SHOWSQL    TRK   20        20        _________ _________ _________ SYSALLDA

Edit XML   TRK   20        20        _________ _________ _________ SYSALLDA


Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

ZDT/Db2 may allocate data sets, depending on function. The allocation attributes for Audit Log, Print, Trace, SHOWSQL (used to display SQL) and XML Edit can be specified here. The Audit Log, Print, and Trace data sets are always permanent (catalogued) data sets. SHOWSQL data sets may be either temporary or permanent. XML Edit data sets are always temporary data sets. Temporary data sets have a high level qualifier like SYSxxxxx and are deleted automatically when the ZDT/Db2 function ends.

Any allocation values specified on this panel might be changed by SMS as part of the allocation process, and temporary data sets might be allocated to VIO, depending on SMS and other system settings.

Space Units

Defines the unit of primary and secondary space to be allocated.

BLK

Block of average size.

KB

Kilobyte, a kilobyte is 1024 bytes.

MB

Megabyte, a megabyte is 1048576 bytes.

TRK

Track of a direct access storage device (DASD).

CYL

Cylinder of a DASD.

Primary Quantity

Amount of DASD space to be used for primary space allocation. The range depends on the space unit specified and the DASD device type.

Secondary Quantity

Amount of DASD space to be used for secondary space allocation. The range depends on the space unit specified and the DASD device type.

Management Class

Name of a management class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Storage Class

Name of a storage class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Data Class

Name of a data class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Device Type

Specifies the device type of the volume to contain the data set. Enter an IBM® device type or a generic device name that is supported by your system.

Parent panels

- Set Processing Options panel on page 832

Child panels

- None.

Comments

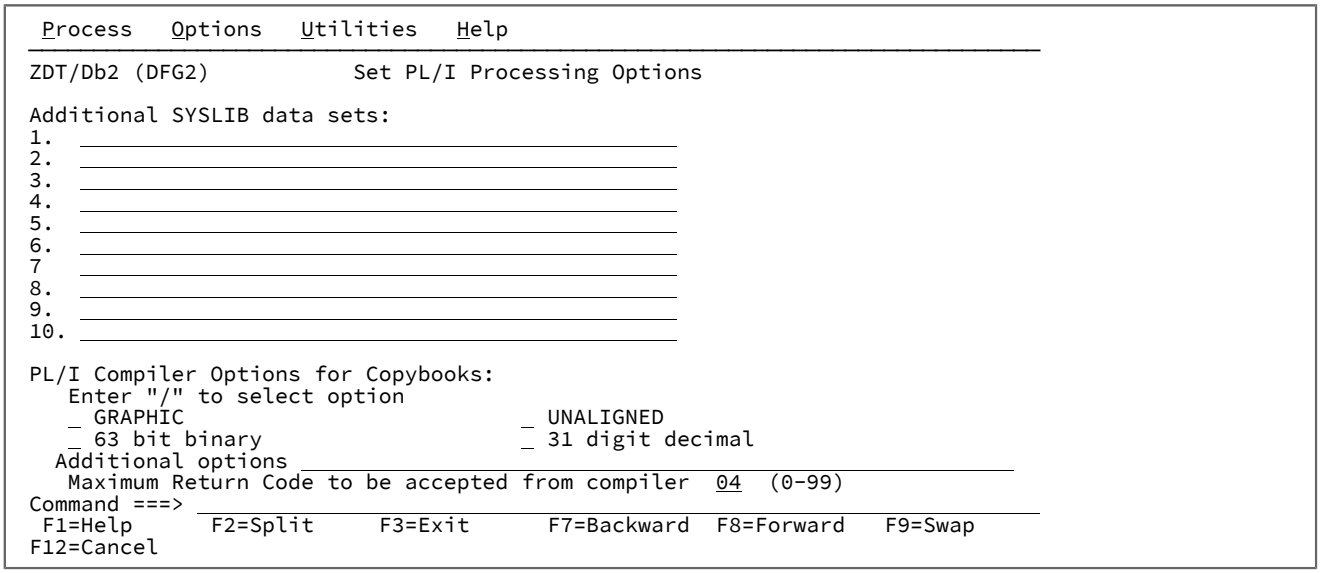
- None.

Set PL/I Processing Options panel

You use the Set PL/I Processing Options panel to customize how ZDT/Db2 compiles PL/I copybooks to generate ZDT/Db2 templates.

These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



Additional SYSLIB data sets

Any additional SYSLIB data sets to be used in the compilation of PL/I copybooks.

PL/I Compiler Options

These options determine how ZDT/Db2 compiles PL/I copybooks to generate ZDT/Db2 templates.

Additional SYSLIB data sets

The names of data sets to be searched if your copybooks contain references to other PL/I copybooks.

GRAPHIC

Select this option if the copybook contains DBCS or mixed data.

UNALIGNED

Select this option if fields are not aligned on full-word boundaries.

63 bit binary

Select this option if you require support for 63 bit binary numbers. The PL/I compiler option LIMITS(FIXEDBIN(63)) is used when you select this option.

31 digit decimal

Select this option if you require support for 31 digit decimal numbers. The PL/I compiler option LIMITS(FIXEDDEC(31)) is used when you select this option.

Additional options

Additional PL/I compiler options which are added via the *PROCESS statement when a PL/I compile is run to build or update template. These options are validated during the compile process. To avoid compile errors ensure that the syntax is correct and that any additional data sets required by these options are allocated before invoking Z Data Tools.

For details on the effects of using these options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

Maximum Return Code

Sets the maximum warning/error code accepted from the compiler when compiling a copybook. Any code higher than specified causes ZDT/Db2 to display a pop-up panel, in which you can review the compilation and decide about future processing.

Child panels

None.

Related tasks

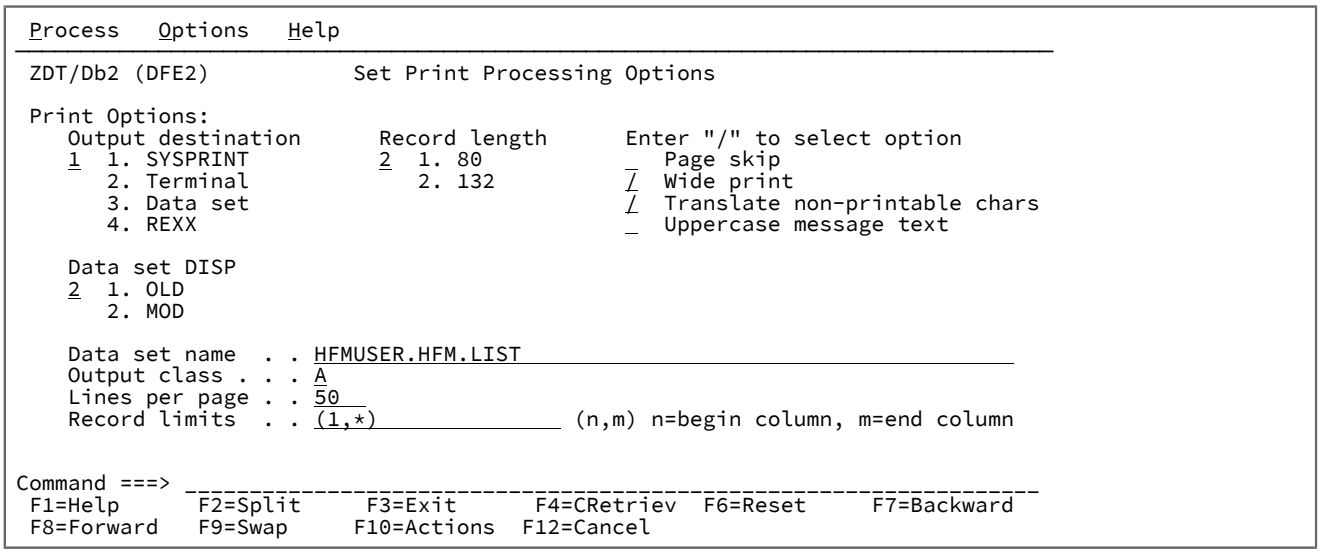
- Setting default processing options on page 62

Set Print Processing Options panel

You use the Set Print Processing Options panel to control processing of the current ZDT/Db2 session.

These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



Output destination

Specifies where print output is sent, except for batch execution.

SYSPRINT

Send print output to the current SYSPRINT allocation.

Terminal

Send print output to the TSO terminal.

Data set

Output is accumulated in the print data set specified in the **Data set name** field. This data set can be browsed using the PB command, or sent for printing to the JES spool queue with class specified in the **Output class** field using the PRINT command while browsing.

REXX

Send output to a REXX stem variable. Each line corresponds to a variable named FILEM.*nnn*. FILEM.0 contains the line count.

When you use the PRINT primary command in Print Browse (option 3.9), or if you specify batch execution in a Z Data Tools/Db2 panel, then print output is always sent to a SYSOUT class.

Record length

Specifies the line length of print output.

80

The line length of print output is 80 characters, suitable for a terminal.

132

The line length of print output is 132 characters, suitable for a printer.

Note:
Record length is not applicable when:

- The data is printed in TABL format, or
- You specify the wide format for the compare output listing

In either of the above cases, when data is printed the line length is determined from the number and size of fields printed. If this length is greater than the record length specified for the data set, the print line is truncated. The maximum print line length is 32760.

Data set DISP

Specifies the disposition of a print data set.

OLD

The print data set is cleared before each print operation, and print output is written from the beginning of the data set.

MOD

Default setting. Print output is appended to the existing print data set.

Page skip

If selected, print output from each function begins on a new page.

Wide print

If selected, for to-be-allocated (new) print output datasets (online) or for SYSPRINT allocation (in batch), ZDT/Db2 uses the maximum record length/blocksize of 32756/32760 and record format of VBA.

If not selected, for to-be-allocated (new) print output datasets (online) or for SYSPRINT allocation (in batch), ZDT/Db2 uses the record length/blocksize of 133/137 and record format of VBA.

Note: This setting only applies when the output dataset is newly-allocated during execution. It is ignored for pre-allocated print output datasets.

Translate non-printable chars

Specifies how ZDT/Db2 should translate non-printable characters.

If selected, non-printable characters are translated to blanks using a translate table.

If not selected, no translation is performed. Use PRTTRANS=OFF to support special print chains. This may make printing faster.

For instructions on altering the print translate table, see the *Z Data Tools Customization Guide*.

Uppercase message text

All message text is translated to uppercase.

Data set name

If you specified the **Output destination** as **3** (Data set), then after you enter the PRINT command while executing the Print Browse (PB) function, ZDT/Db2 sends print output to this temporary data set.

The default is *userid*.HFM.LIST, unless changed in your ZDT/Db2 installation.

Output class

The class of the JES spool queue to be used when the PRINT command is issued while browsing the temporary print data set.

Lines per page

Specifies the number of lines printed on each page of print output. You can specify a value from 1 to 999. The default is 60.

Record limits

When data is printed in CHAR or LHEX print format, limits print output for each record (or OAM object).

(1,*)

The entire record (or block) is printed.

(*n*,*m*)

Print output is limited to the data from columns (bytes) *n* through *m*. An asterisk (*) specified for *m* indicates the end of the record.

Parent panels

- Set System Processing Options panel on page 835

Child panels

None.

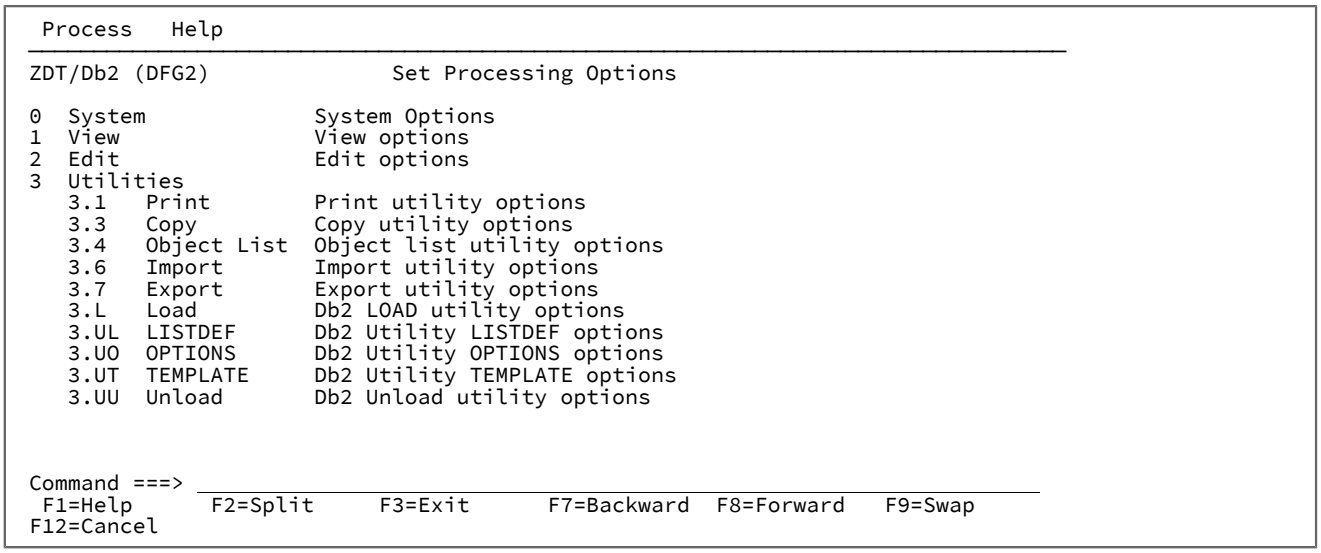
Related tasks

- Print settings (option 0.1) on page 64
- PRINT primary command on page 989
- Looking at the print output from your ZDT/Db2 session on page 333

Set Processing Options panel

You use the Set Processing Options panel to navigate to other panels from which you can set various types of options that affect the behavior of ZDT/Db2 panels. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions



0 System Options

Displays the Set System Processing Options panel

1 View

Displays the first Editor Options panel

2 Edit

Displays the first Editor Options panel

3 Utilities

Displays the Set Utilities Processing Options panel

3.1 Print

Displays the Print Utility Options panel

3.3 Copy

Displays the Copy Options panel

3.4 Object List

Displays the ZDT/Db2 Object List Options (1 of 2) panel

3.6 Import

Displays the Import Options panel

3.7 Export

Displays the first Export Options panel

3.L LOAD

Displays the Load Utility Options panel

3.UL LISTDEF

Displays the LISTDEF Options panel

3.UO OPTIONS

Displays the OPTIONS Options panel

3.UT TEMPLATE

Displays the TEMPLATE Options (1 of 2) panel

3.UU Unload

Displays the UNLOAD Options panel

Parent panels

- Primary Option Menu panel on page 760

Child panels

To display this panel...	Use/do this
Set System Processing Options panel on page 835	Option 0
Set Print Processing Options panel on page 827	Option 0.1
Systems Options (1 of 4) panel on page 857	Option 0.2
Set Batch Job Card Information panel on page 817	Option 0.3
Compiler Language Selection panel on page 506	Option 0.4
Set COBOL Processing Options panel on page 818	Option 0.5
Set HLASM Processing Options panel on page 821	Option 0.6
Set PL/I Processing Options panel on page 825	Option 0.7
Set Temporary Data Set Allocation Options panel on page 837	Option 0.8
Set Output Data Set Allocation Options panel on page 824	Option 0.9
ISPF Settings panel on page 727	Option 0.10
Editor Options (1 of 8) panel on page 644	Option 1 or 2

To display this panel...	Use/do this
Set Utility Options panel on page 840	Option 3
Print Utility Options panel on page 763	Option 3.1
Copy Options panel on page 509	Option 3.3
Object List Options (1 of 2) panel on page 743	Option 3.4
Import Options panel on page 716	Option 3.6
Export Options (1 of 3) panel on page 689	Option 3.7
LOAD Utility Options panel on page 737	Option 3.L
LISTDEF Options panel on page 729	Option 3.UL
OPTIONS Options panel on page 750	Option 3.UO
TEMPLATE Options (1 of 2) panel on page 897	Option 3.UT
UNLOAD Options panel on page 905	Option 3.UU

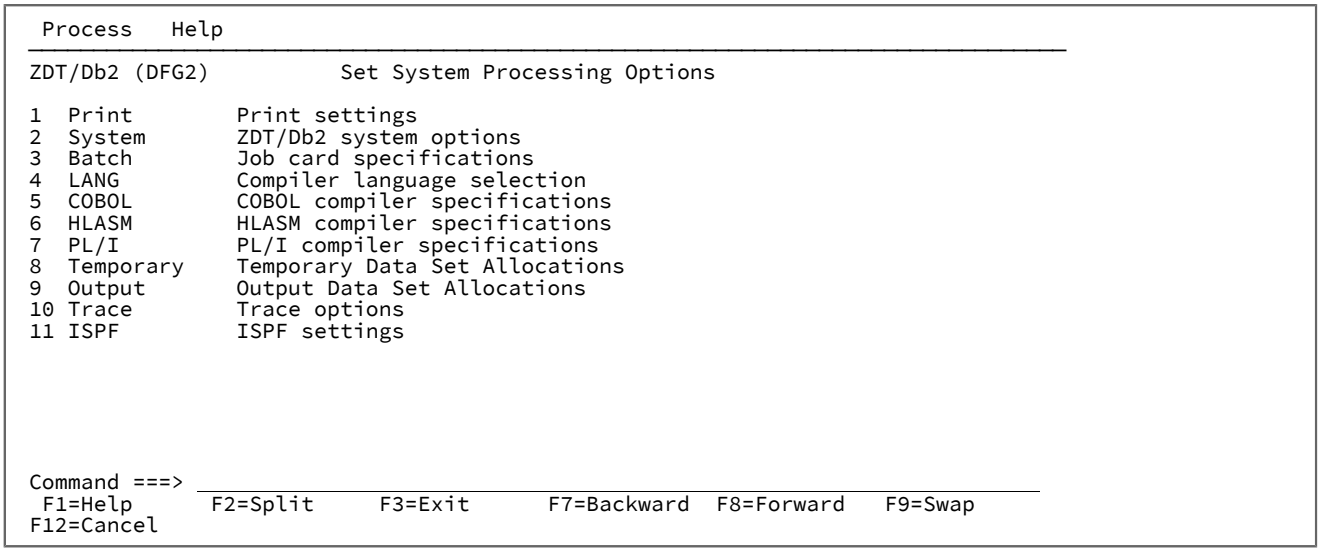
Related tasks

- Setting default processing options on page 62

Set System Processing Options panel

You use the Set System Processing Options panel to navigate to other options panels.

Panel and field definitions



1 Print

Displays the Set Print Processing Options panel

2 System

Displays the first ZDT/Db2 System Options panel

3 Batch

Displays the Set Batch Job Card Information panel

4 LANG

Displays the Compiler Language Selection panel

5 COBOL

Displays the Set COBOL Processing Options panel

6 HLASM

Displays the Set HLASM Processing Options panel

7 PL/I

Displays the Set PL/I Processing Options panel

8 Temporary

Displays the Set Temporary Data Set Allocation Options panel

9 Output

Displays the Set Output Data Set Allocation Options panel

10 Trace

Displays the Set Trace options panel

11 ISPF

Displays the ISPF Settings panel

For a description of the fields on the ISPF Settings panel, see the relevant *z/OS ISPF User's Guide Vol II*.

Parent panels

- Set Processing Options panel on page 832

Child panels

To display this panel...	Use/do this
Set Print Processing Options panel on page 827	Option 1
Systems Options (1 of 4) panel on page 857	Option 2
Set Batch Job Card Information panel on page 817	Option 3
Compiler Language Selection panel on page 506	Option 4
Set COBOL Processing Options panel on page 818	Option 5
Set HLASM Processing Options panel on page 821	Option 6
Set PL/I Processing Options panel on page 825	Option 7
Set Temporary Data Set Allocation Options panel on page 837	Option 8
Set Output Data Set Allocation Options panel on page 824	Option 9
Set Trace options panel on page 839	Option 10
ISPF Settings panel on page 727	Option 11

Related tasks

- Setting default processing options on page 62

Set Temporary Data Set Allocation Options panel

Panel and field definitions

Figure 133. Set Temporary Data Set Allocation Options panel

```
 Process   Options   Help
-------------------------------------------------------------------------------
ZDT/Db2 (DFG2)    Set Temporary Data Set Allocation Options

Set processing options as desired and enter EXIT (F3) to save your changes.
Enter RESET to restore installation defaults.

Temporary data set allocation defaults:
   Unit . . . . . SYSALLDA                leave blank for default
   Data class . . . . . _________         leave blank for default
   Storage class  . . . _________         leave blank for default
   Management class . . _________         leave blank for default

Default High Level Qualifiers:
   Temporary Data Sets  ____________________  HLQ (opt. &USER/&PREFIX)




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Temporary data set allocation defaults

Unit

Can be one of these:

- A 3-digit hexadecimal device number.
- A 4-digit hexadecimal device number prefixed with a slash (/).
- Device type (generic name).
- Group name (installation-defined symbolic name).

Data class

Name of a data class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Storage class

Name of a storage class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Management class

Name of a management class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Default High Level Qualifiers

Temporary Data Sets

Can be any multi-level qualifier, which is used by Z Data Tools as the high-level qualifier (HLQ) in names of created datasets. You can use these symbols as part of the HLQ:

&USER

Represents the user ID.

&PREFIX

Represents the TSO prefix.

The default is none (unless changed in your Z Data Tools installation).

Parent panels

- Set Processing Options panel on page 832

Child panels

- None.

Comments

- None.

Set Trace options panel

The Set Trace options panel allows you to view and modify parameters that control the Trace output produced by ZDT/Db2 when it is run in debug mode.

Panel and field definitions

Figure 134. Set Trace options panel

```
 Process   Options   Help
--------------------------------------------------------------------------------
ZDT/Db2 (DFE2)               Set Trace options

Trace Options:
   Trace destination
   3  1. HFMTRC
      2. Terminal
      3. Data set

   Data set name  JHELVON.HFM.TRACE
   Output class . A
   Trace limit .  500




Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

Trace destination

Specifies where debug trace ZDT/Db2 output is to be sent:

HFMTRC

Send trace output to the current HFMTRC allocation.

Terminal

Send trace output to the terminal.

Data set

Send trace output to the temporary data set specified by the **Data set name** field.

Data set name

The temporary trace data set where ZDT/Db2 trace output is directed when the Data set trace destination is specified.

Output class

The class of the JES spool queue to be used when the PRINT command is issued when browsing the temporary trace data set.

Trace limit

When run in debug mode, ZDT/Db2 writes the Trace output to a buffer (as for normal debug mode), but the buffer is only written to the Trace destination, if ZDT/Db2 abnormally terminates.

This field specifies the number of Trace statements that are retained in the buffer. If you specify a value of zero, background debug mode is turned off.

Parent panels

Set System Processing Options panel on page 835

Child panels

None.

Related tasks

Set Utility Options panel

You use the Set Utility Processing Options panel to navigate to other options panels.

Panel and field definitions

```
 Process   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)          Set Utility Processing Options

1   Print        Print utility options
3   Copy         Copy utility options
4   Object list  Object list utility options
6   Import       Import utility options
7   Export       Export utility options
L   LOAD         Db2 LOAD utility options
UL  LISTDEF      Db2 Utility LISTDEF options
UO  OPTIONS      Db2 Utility OPTIONS options
UT  TEMPLATE     Db2 Utility TEMPLATE options
UU  Unload       Db2 Unload utility options




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

1 Print

Displays the Copy Options panel

3 Copy

Displays the Copy Options panel

4 Object list

Displays the ZDT/Db2 Object List Options (1 of 2) panel

6 Import

Displays the Import Options panel

7 Export

Displays the first Export Options panel

L LOAD

Displays the Load Utility Options panel

UL LISTDEF

Displays the LISTDEF Options panel

UO OPTIONS

Displays the OPTIONS Options panel

UT TEMPLATE

Displays the TEMPLATE Options (1 of 2) panel

UU Unload

Displays the UNLOAD Options panel

Parent panels

- Set Processing Options panel on page 832

Child panels

To display this panel…	Use/do this
Copy Options panel on page 509	Option 3
Object List Options (1 of 2) panel on page 743	Option 4
Import Options panel on page 716	Option 6
Export Options (1 of 3) panel on page 689	Option 7
LOAD Utility Options panel on page 737	Option L
LISTDEF Options panel on page 729	Option UL
OPTIONS Options panel on page 750	Option UO
TEMPLATE Options (1 of 2) panel on page 897	Option UT
UNLOAD Options panel on page 905	Option UU

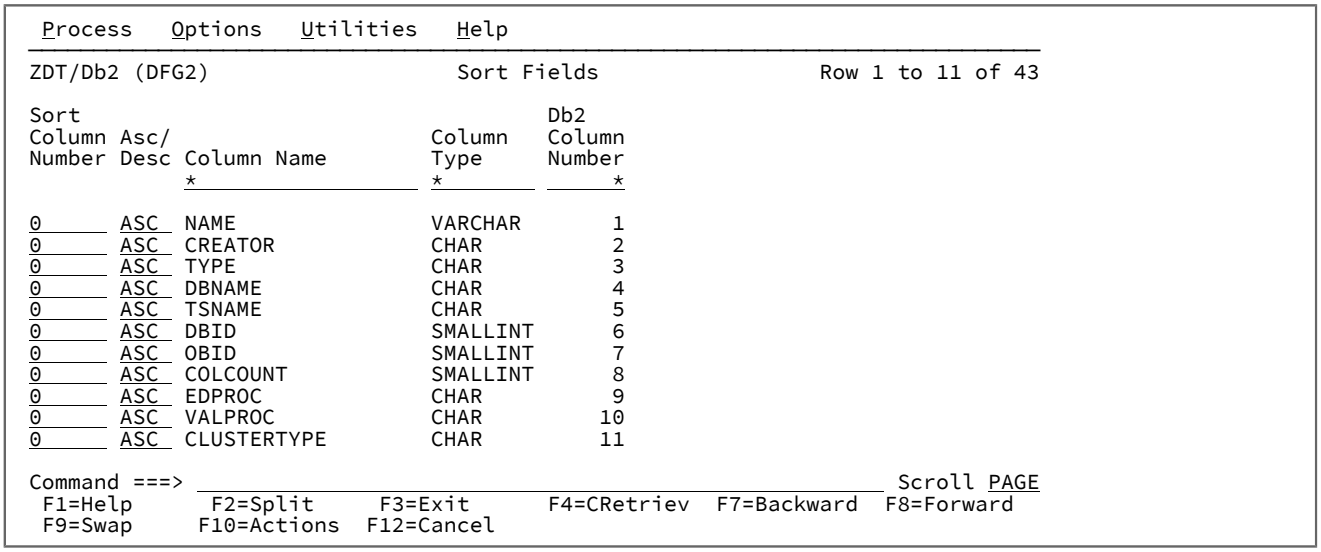
Related tasks

- Setting default processing options on page 62

Sort Fields panel

You use the Sort Fields panel to sort multiple columns within the current object list.

Panel and field definitions



```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Sort Fields                Row 1 to 11 of 43

Sort                                          Db2
Column Asc/                      Column       Column
Number Desc Column Name          Type         Number
            *                    *            *

0      ASC  NAME                 VARCHAR           1
0      ASC  CREATOR              CHAR              2
0      ASC  TYPE                 CHAR              3
0      ASC  DBNAME               CHAR              4
0      ASC  TSNAME               CHAR              5
0      ASC  DBID                 SMALLINT          6
0      ASC  OBID                 SMALLINT          7
0      ASC  COLCOUNT             SMALLINT          8
0      ASC  EDPROC               CHAR              9
0      ASC  VALPROC              CHAR             10
0      ASC  CLUSTERTYPE          CHAR             11

Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Sort Column Number

The relative position of this column in the sort sequence. Specify 1 for the most significant column, 2 for the next significant, and so on.

Asc/Desc

Indicates if you want ZDT/Db2 to sort the column in ascending or descending sequence:

ASC

Ascending. To change from DESC, overtype with ASC or A.

DESC

Descending. To change from ASC, overtype with DESC or D.

Column Name

The name of the column as held in the Db2® catalog table.

Column Type

The data type of the column.

Db2® Column Number

The number of the column as held in the Db2® catalog table.

Parent panels

The Sort Fields panel is displayed when you enter `SORT` on the command line from any object list and the cursor is *not* positioned within any column.

Child panels

To display this panel...	Use/do this
Databases panel on page 609	Object Type 1
Table Spaces panel on page 879	Object Type 2
Tables, Views and Aliases panel on page 891	Object Type 3
Indexes panel on page 723	Object Type 4
Columns panel on page 503	Object Type 5
Synonyms panel on page 855	Object Type 6
Application Plans panel on page 473	Object Type 7
Application Packages panel on page 469	Object Type 8
Collections panel on page 482	Object Type 9
Database Request Modules panel on page 607	Object Type 10
Schemas panel on page 812	Object Type 11
Distinct Types panel on page 637	Object Type 12
Functions panel on page 709	Object Type 13
Stored Procedures panel on page 852	Object Type 14
Triggers panel on page 903	Object Type 15

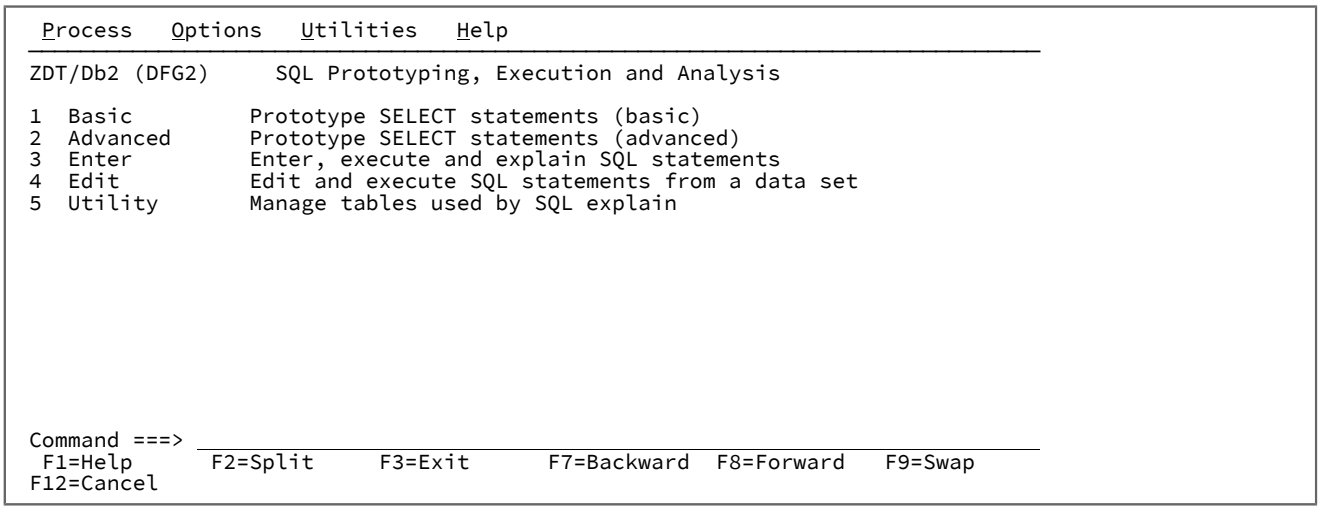
Related tasks

- Sorting the displayed data on page 323

SQL Prototyping, Execution and Analysis panel

You use the SQL Prototyping, Execution and Analysis panel to select the ZDT/Db2 SQL function you want to use.

Panel and field definitions



1 (Basic)

Use this option to build and execute a basic SELECT statement.

2 (Advanced)

Use this option to build and execute a complex SELECT statement.

3 (Enter)

Use this option to enter, execute, and explain SQL statements.

4 (Edit)

Use this option to edit and execute SQL statements from a data set.

5 (Utility)

Use this option to select utilities to support the use of the Db2® EXPLAIN statement.

Parent panels

- Primary Option Menu panel on page 760

Child panels

To display this panel...	Use/do this
Basic SELECT Prototyping panel on page 476	Option 1
Advanced SELECT Prototyping panel on page 451	Option 2
Enter, Execute and Explain SQL Statements panel on page 684	Option 3
Edit/Execute SQL (Data Set) panel on page 641	Option 4
Explain Utilities panel on page 686	Option 5

Related tasks

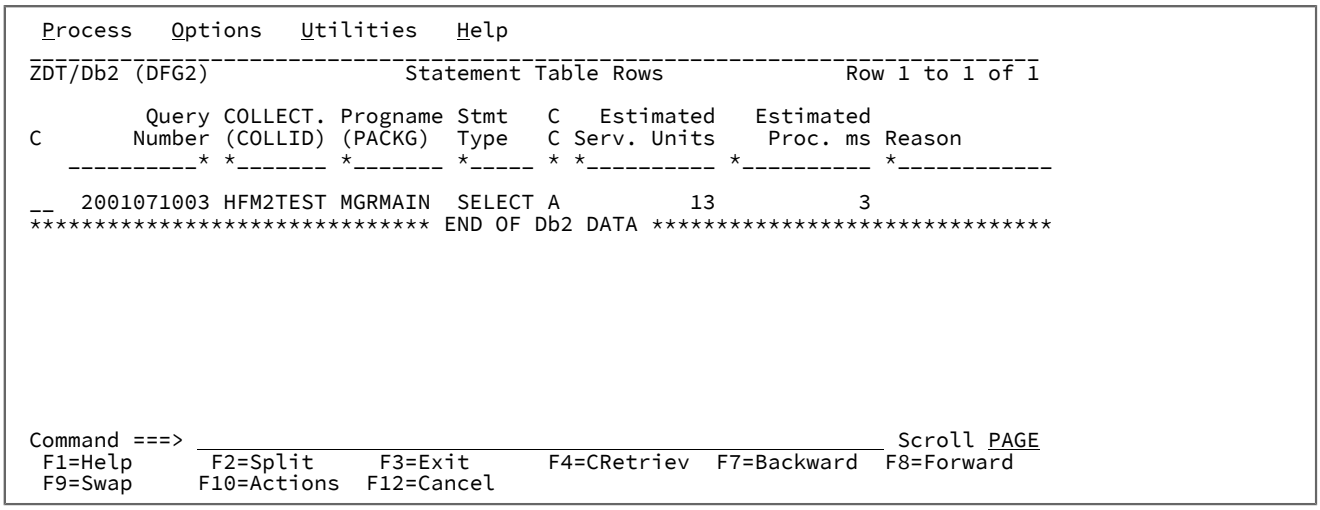
- Using basic SQL prototyping on page 390
- Using advanced SQL prototyping on page 399
- Entering, executing, and explaining SQL statements on page 424
- Editing and executing SQL statements from a data set on page 429
- Managing tables used by SQL explain on page 430

Statement Table Rows panel

ZDT/Db2 displays the Statement Table Rows panel if you use the SR line command against an item listed on the Plan Table Rows panel on page 754.

Use this panel to display the rows from a table.

Figure 135. Statement Table Rows panel (collection ID format)



The Statement Table Rows panel shows you rows from a statement table:

C

Input field where you can enter one of the following line commands:

DK

Delete rows for package

DP

Delete rows for plan

DQ

Delete rows for query number

I

Interpret EXPLAIN data

K

Show package

M

Show DBRM

P

Show plan

Query Numb

A number that identifies the SQL statement.

Applname

PLAN name or blank for dynamic EXPLAIN. This information only is displayed for the PLAN format of the panel.

Collect.

Collection name. This information only is displayed for the COL format of the panel.

Progname

DBRM/package name.

Stmt Type

Statement type.

C C

Cost category:

A

Not using default values.

B

Using default values when calculating costs.

Est. SU

Estimated processor cost in service units.

Est. P ms.

Estimated processor cost in milliseconds.

Reason

A string indicating the reason for putting an estimate into cost category B (see **C C**).

Parent panels

- Plan Table Rows panel on page 754

Child panels

To display this panel...	Use/do this
EXPLAIN Interpretation panel on page 686	Line command I
Application Packages panel on page 469	Line command K

Related tasks

- Managing tables used by SQL explain on page 430

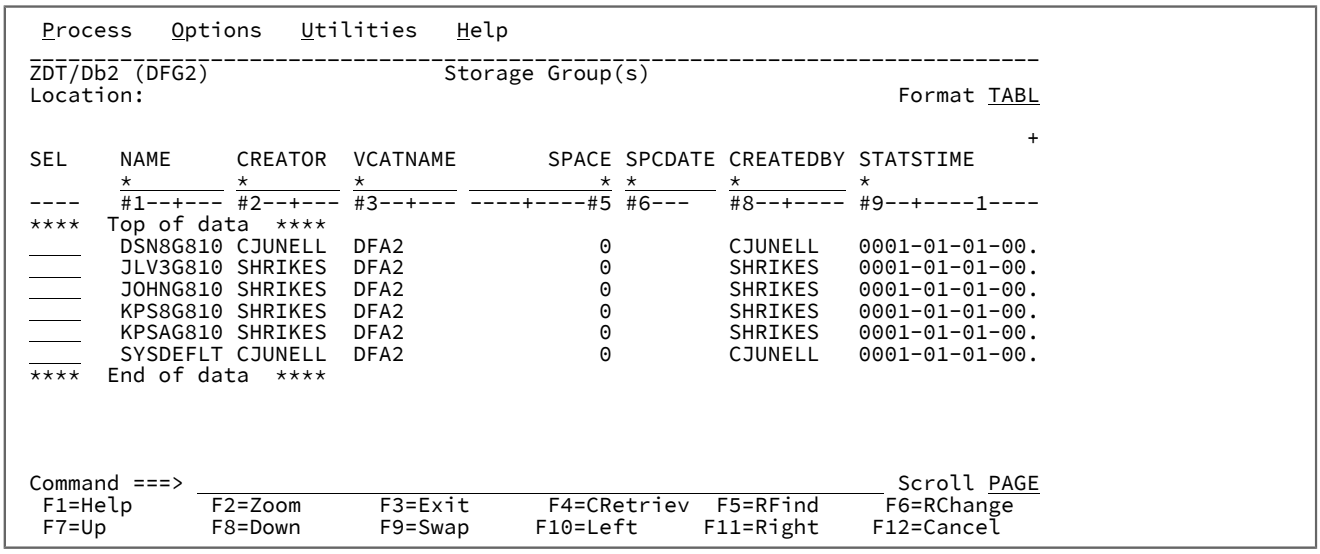
Storage Group(s) panel

You use the Stored Group(s) panel to display information about the storage groups you have selected.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSSTOGROUP.

Figure 136. Storage Group(s) panel



Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748

Child panels

To display this panel...	Use/do this
	Line command D

To display this panel...	Use/do this
Databases panel on page 609	
	Line command P
Storage Group Usage Privileges panel on page 851	
	Line command PT
Table Space Parts panel on page 879	
	Line command ROW
Display Row panel on page 637	
	Line command S
Table Spaces panel on page 879	
	Line command VOL
Volumes panel on page 919	
	Line command X
Indexes panel on page 723	
	Primary command XPT
Index Parts panel on page 723	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Related tasks

- Managing tables used by SQL explain on page 430

Storage Group Usage Privileges panel

Figure 137. Storage Group Usage Privileges panel

```
 Process   Options   Utilities   Help
_______________________________________________________________________________
ZDT/Db2 (DFG2)    Storage Group Usage Privileges for DSN8G810
                                                                  Format TABL


SEL    GRANTOR  GRANTEE  QUALIFIER NAME     AUTHHOEX OBTYPEEX      USEAUTEX
       *_______ *_______ *_______ *_______ *_______ *_____________ *_______
----   #1--+--- #2--+--- #3--+---- #4--+--- #15-+--- #16-+----1--- #17--
****  Top of data  ****
_____   CJUNELL  CJUNELL           DSN8G810 SYSADM   Storage Group Grant
_____   CJUNELL  PUBLIC            DSN8G810 SYSADM   Storage Group Yes
_____   SHRIKE2  ID1               DSN8G810 SYSADM   Storage Group Yes
****  End of data  ****









Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Storage Group(s) panel on page 848

Child panels

To display this panel…	Use/do this
	Line command G
Grant Storage Group Privileges panel	
	Line command I

To display this panel...	Use/do this
Storage Group Use Privilege Details panel	
	Line command R
Revoke Storage Group Privileges panel	
	Line command ROW
Display Row panel on page 637	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related tasks

- Managing tables used by SQL explain on page 430

Stored Procedures panel

You use the Stored Procedures panel to display information about the stored procedures you have selected.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSROUTINES.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                Stored Procedures
Location:                                                      Format TABL

SEL    SCHEMA   NAME                ORIGINEX OWNER    ROUTINEX          CREATEDB
       *        *                   *        *        *                 *
----   #1--+--- #3--+----1----+--- #90-+--- #2--+--- #89-+----1----+- #5--+---
****  Top of data  ****
_____  SYSPROC  DSNWZP              External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  DSNWSPM             External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  DSNUTILS            External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  DSNUTILU            External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  DSNTPSMP            External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  WLM_REFRESH         External CJUNELL  Stored procedure CJUNELL
_____  SQLJ     INSTALL_JAR         External CJUNELL  Stored procedure CJUNELL
_____  SQLJ     REPLACE_JAR         External CJUNELL  Stored procedure CJUNELL
_____  SQLJ     REMOVE_JAR          External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  DSNACICS            External CJUNELL  Stored procedure CJUNELL
_____  SYSPROC  DSNACCOR            External CJUNELL  Stored procedure CJUNELL
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:



1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Schemas panel on page 812

Child panels

To display this panel…	Use/do this
	Line command A
Alter Stored Procedure panel on page 456	
	Line command COM
See Comment panels on page 506	
	Line command CR
Create Procedure panel on page 553	
	Line command DR
See Drop panels on page 640	
	Line command G
Grant privileges panels on page 715	
	Line command GEN

To display this panel...	Use/do this
Generate SQL From Db2 Catalog panel on page 712	
	Line command I
Details panels on page 636	
	Line command P
Privileges panels on page 768	
	Line command PARM
Routine Parameters panel on page 799	
	Line command PKG
Application Packages panel on page 469	
	Line command PS
	Line command R
Revoke privileges panels on page 798	
	Line command ROW
Display Row panel on page 637	
	Line command SCH
Schemas panel on page 812	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Synonym Dependencies panel

ZDT/Db2 displays the Synonym Dependencies panel if you use the SYN line command against an item listed on:

- The Application Plans panel on page 473, or
- The Application Packages panel on page 469

Synonyms panel

You use the Synonyms panel to list synonyms in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSSYNONYMS.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Synonyms
Location:                                                    Format TABL

SEL    CREATOR  NAME                TBCREATOR    TBNAME              CREATEDBY
       *        *                   *            *                   *
----   #2--+--- #1--+----1----+--- #4--+----1-- #3--+----1----+--- #6--+----
****  Top of data  ****
_____  CJUNELL  ACT                 DSN8810      ACT                 CJUNELL
_____  CJUNELL  DEPT                DSN8810      DEPT                CJUNELL
_____  CJUNELL  EMP                 DSN8810      EMP                 CJUNELL
_____  CJUNELL  EMPPROJACT          DSN8810      EMPPROJACT          CJUNELL
_____  CJUNELL  PROJ                DSN8810      PROJ                CJUNELL
_____  CJUNELL  PROJACT             DSN8810      PROJACT             CJUNELL
_____  CJUNELL  TCONA               DSN8810      TCONA               CJUNELL
_____  CJUNELL  TDSPTXT             DSN8810      TDSPTXT             CJUNELL
_____  CJUNELL  TOPTVAL             DSN8810      TOPTVAL             CJUNELL
_____  CJUNELL  VACT                DSN8810      VACT                CJUNELL
_____  CJUNELL  VASTRDE1            DSN8810      VASTRDE1            CJUNELL
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Tables, Views and Aliases panel on page 891

Child panels

To display this panel…	Use/do this
Create Synonym panel on page 560	Line command CR
Create Alias panel on page 527	Line command CRA
See Drop panels on page 640	Line command DR
Details panels on page 636	Line command I
Display Row panel on page 637	Line command ROW
Tables, Views and Aliases panel on page 891	Line command T
Sort Fields panel on page 841	Primary command SORT

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Systems Options (1 of 4) panel

You use the ZDT/Db2 Systems Options (1 of 4) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 138. ZDT/Db2 System Options (1 of 4) panel

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)         ZDT/Db2 System Options (1 of 4)          Global Settings

Uppercase Translation:
   Enter "/" to select option
   /  Translate Db2 object names
   /  Translate input SQL statements

Encapsulation of SQL Identifiers in Double Quotes:
   Use double quotes
   3  1. For Db2 SQL reserved words
      2. For IBM and Db2 SQL reserved words
      3. Always

More Options:
   Enter "/" to select option
   /  Arbitrary select statements, time display format, template creation
   /  CCSID warning option
   /  Editor choice for viewing output data, members; cancel confirmation

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F11=NxtPage  F12=Cancel
```

Translate Db2® object names

Translates Db2® object owner and Db2® object name to uppercase. (The Db2® object location is always translated to uppercase.) The default is selected.

If you do not select this option, the Db2® object owner and Db2® object name remain as entered by the user.

Translate input SQL statements

Translates any free-form SQL statements entered to uppercase. The default is selected.

If you do not select this option, any free-form SQL statements entered remain as entered by the user.

Encapsulation of SQL Identifiers in Double Quotes

The SQL identifiers encapsulated in double quotation marks in SQL statements generated by ZDT/Db2. You can select one of the following:

- **1. For Db2® SQL reserved words**

  If your Db2® objects include column names that are Db2® SQL reserved words, you must enclose these column names in double quotation marks. Use this option to only enclose identifiers that are Db2® SQL reserved words in double quotation marks. Any other identifiers appear in SQL statements without encapsulating double quotation marks. This option is recommended if you are using the SQL primary command (see SQL primary command on page 1005) to generate and save SQL statements for later use.
- **2. For IBM® and Db2® SQL reserved words**

  Use this option to recognize any identifiers that are either a Db2® SQL reserved word, or an IBM® SQL reserved word, and enclose these words in double quotation marks. Any other identifiers appear in SQL statements without encapsulating double quotation marks.
- **3. Always**. This is the default setting. This option applies to view, edit, browse, print, copy, import and export functions only and results in all identifiers being enclosed in double quotation marks in SQL statements generated by ZDT/Db2.

Note:

1. For information about the rules covering the construction of SQL identifiers, see the *DB2 for z/OS SQL Reference*.
2. In general, you can use any uppercase alphabetic character and three special characters ($, _, and @) in an SQL identifier without special consideration. If your identifier includes lowercase characters, spaces, or other characters, then you must enclose the identifier in double quotation marks. ZDT/Db2 detects identifiers with lowercase and other unusual characters and encloses these in double quotation marks. This occurs regardless of the encapsulation option selected.
3. The lists of Db2® and IBM® SQL reserved words used by ZDT/Db2 are the latest available. When connected to an older release of Db2®, if you have selected either options 1 or 2, ZDT/Db2 might recognize an identifier as a reserved word, even though the identifier is not defined as a reserved word in the earlier release of Db2®.

Parent panels

- Set System Processing Options panel on page 835

Child panels

To display this panel…	Use/do this
Systems Options (2 of 4) panel on page 859	Press the NxtPage function key (F11)

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- TIME columns on page 168
- INDEXINF primary command on page 970
- “No Primary Key” errors on page 203
- “Row Not Found” errors on page 205

Related references

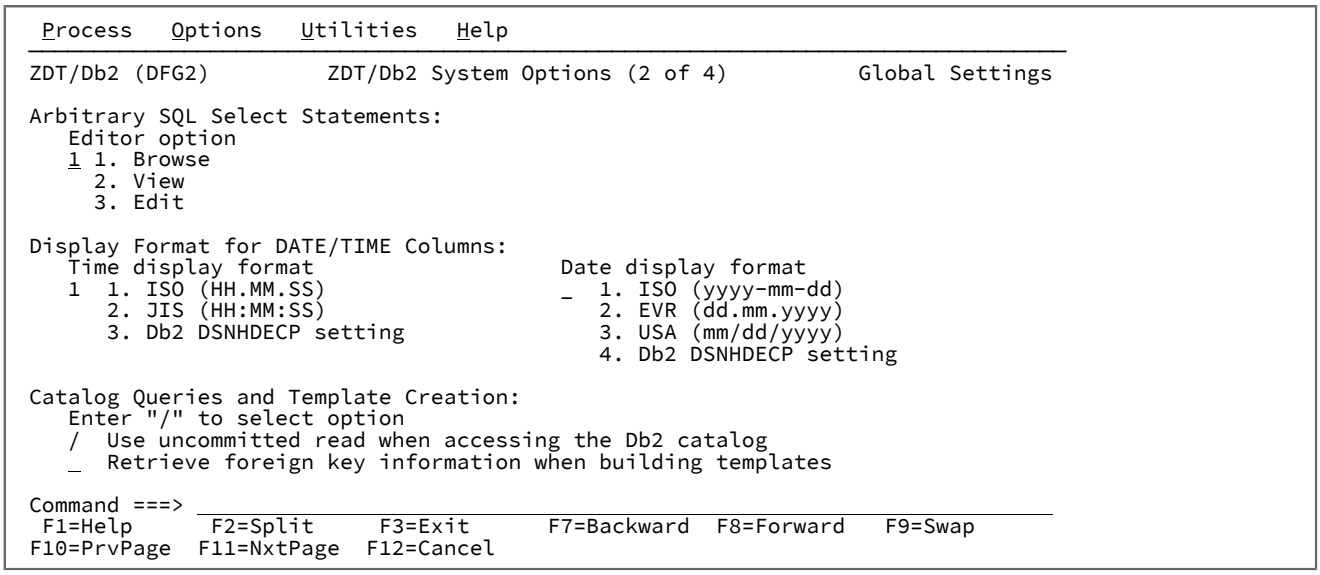
- SQL primary command on page 1005

Systems Options (2 of 4) panel

You use the ZDT/Db2 Systems Options (2 of 4) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 139. ZDT/Db2 System Options (2 of 4) panel



Arbitrary SQL Select Statements

The type of editor that is used when the result table is displayed. The result table is generated by a freeform SQL statement. The default is 1 (**Browse**).

You can select one of the following options:

Browse

ZDT/Db2 uses **Browse** to display the result table. No prefix area is displayed and you cannot change the data.

View

ZDT/Db2 uses **View** to display the result table. You can change the data by using editor commands, or by over-typing data. However, the changes cannot be saved to Db2®.

Edit

ZDT/Db2 uses **Edit** to display the result table. You can change the data by using editor commands, or by over-typing data. You can possibly save the changes to Db2®. If you choose to disable this option when you install the product, this option might not be available.

Display Format for DATE/TIME Columns

Time display format

The external format for data retrieved from Db2® TIME data type columns. The default for this option is '1' (**ISO**). You can select one of the following:

- **1. ISO** (HH.MM.SS). This is the default format.
- **2. JIS** (HH:MM:SS)
- **3. Db2 DSNHDECP setting** (uses the external format set in the Db2® DSNHDECP module).

Depending on the time format used, information can be lost when data is retrieved from Db2® (for example, the USA format, which discards the seconds portion of any time). If LOCAL is specified (a time format exit is in use), ensure that there is no loss of information when the time value is converted to an external format. When you use this option, you can encounter:

- SQLCODE 100 (row not found) conditions when attempting to delete or update tables that contain a TIME column.
- Removal of the seconds portion of any TIME column (the seconds value is set to 00) in print, copy, or export operations.

Date display format

The external format for data retrieved from Db2® DATE data type columns. The default for this option is '1' (**ISO**). You can select one of the following:

- **1. ISO** (yyyy-mm-dd). This is the default format.
- **2. EUR** (dd.mm.yyyy)
- **3. USA** (mm/dd/yyyy)
- **4. Db2 DSNHDECP setting** (uses the external format set in the Db2® DSNHDECP module).

This setting applies to data displayed in the editor.

Use uncommitted read when accessing the Db2® catalog

This option ensures that all Db2® catalog accesses for the view, edit, print, copy, export and import functions use the uncommitted read option (that is, no locks are taken against the Db2® catalog). The default is selected.

Retrieve foreign key information when building templates

ZDT/Db2 stores information about primary keys, indexes and foreign keys in the template for an object. This information is retrieved from the Db2® catalog and stored at template creation time, and can be displayed in a Z Data Tools/Db2 editor session. Retrieval of the foreign key information can result in slow response if an additional index has not been built on the SYSIBM.SYSFOREIGNKEYS catalog table (see the *Z Data Tools Customization Guide* for details). The default for this option is not selected. When this option is not selected, any templates generated do not contain any foreign key information.

Parent panels

- Systems Options (1 of 4) panel on page 857

Child panels

To display this panel…	Use/do this
Systems Options (1 of 4) panel on page 857	Press the PrvPage function key (F10)
Systems Options (3 of 4) panel on page 862	Press the NxtPage function key (F11)

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65
- TIME columns on page 168
- INDEXINF primary command on page 970
- "No Primary Key" errors on page 203
- "Row Not Found" errors on page 205

Related references

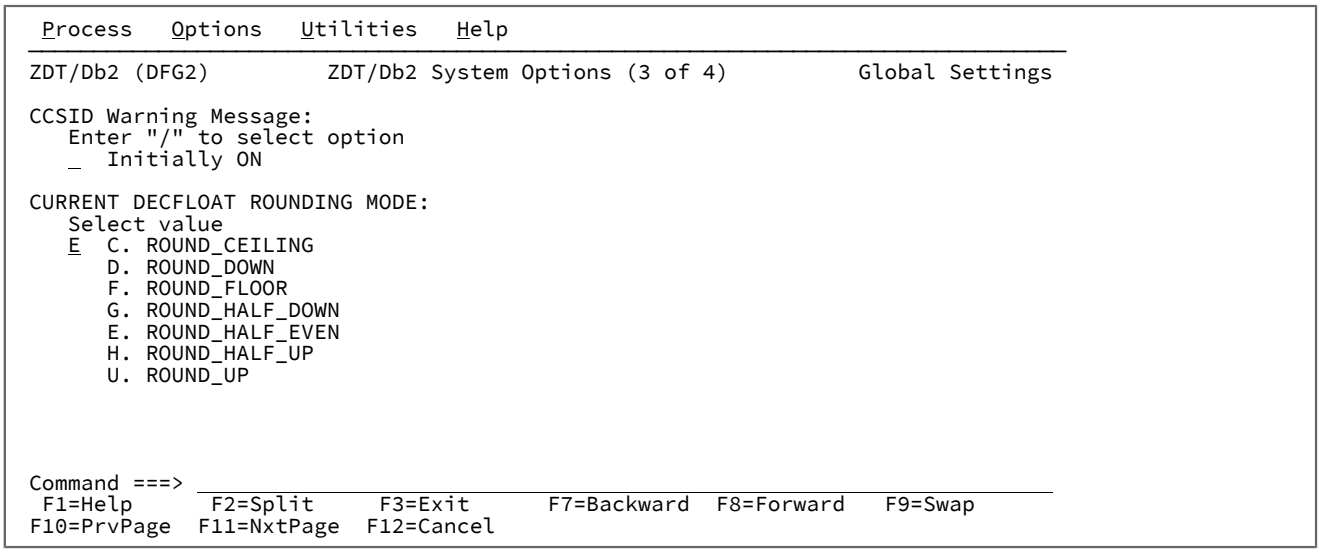
- SQL primary command on page 1005

Systems Options (3 of 4) panel

You use the ZDT/Db2 Systems Options (3 of 4) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 140. ZDT/Db2 System Options (3 of 4) panel



CCSID Warning Message

The setting for this option may have been fixed by the product installer. When this option is available for update, the default is not selected.

You can set your preferred initial system CCSID Warning message option by changing this option. To request a warning message to be displayed when the CCSID of the terminal differs from the CCSID of the plan, select the option. To suppress the displaying of a warning message when the CCSID of the terminal differs from the CCSID of the plan, do not select the option.

CURRENT DECFLOAT ROUNDING MODE

Sets the current decimal floating point rounding mode for the currently connected Db2® system. The default for this option is **E**.

For a detailed explanation of the various decimal floating point rounding modes you can specify, see the *Db2® SQL Reference*.

Setting the option causes ZDT/Db2 to issue an SQL statement that changes the CURRENT DECFLOAT ROUNDING MODE special register.

Parent panels

- Systems Options (2 of 4) panel on page 859

Child panels

To display this panel…	Use/do this
Systems Options (2 of 4) panel on page 859	Press the PrvPage function key (F10)
Systems Options (4 of 4) panel on page 864	Press the NxtPage function key (F11)

Related tasks

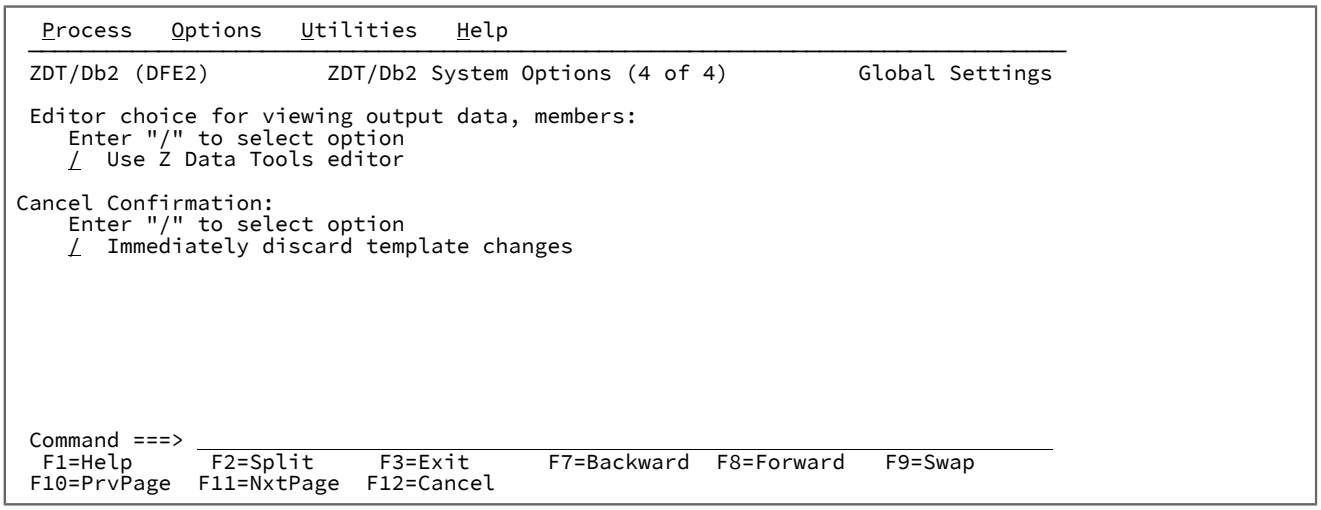
- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65

Systems Options (4 of 4) panel

You use the ZDT/Db2 Systems Options (4 of 4) panel to set various options that affect the way in which ZDT/Db2 operates. These options are saved in your ISPF profile and persist from one ZDT/Db2 session to the next.

Panel and field definitions

Figure 141. ZDT/Db2 System Options (4 of 4) panel



Use Z Data Tools editor

This option controls which editor is invoked when viewing or browsing a member from a member selection panel or when viewing generated output. Select this option to invoke the Z Data Tools editor. If the option is not selected, the ISPF Browse, View or Edit facility is invoked when available. The default for this option is selected.

Cancel confirmation

This option controls whether a warning panel is displayed when you change a template and issue a cancel command. The default is not selected.

When you select this option, unsaved changes to a template are discarded without a warning panel being displayed.

Child panels

To display this panel…	Use/do this
Systems Options (3 of 4) panel on page 862	Press the PrvPage function key (F10)

Related tasks

- Setting default processing options on page 62
- ZDT/Db2 system options (option 0.2) on page 65

Table Browse panel

Panel and field definitions

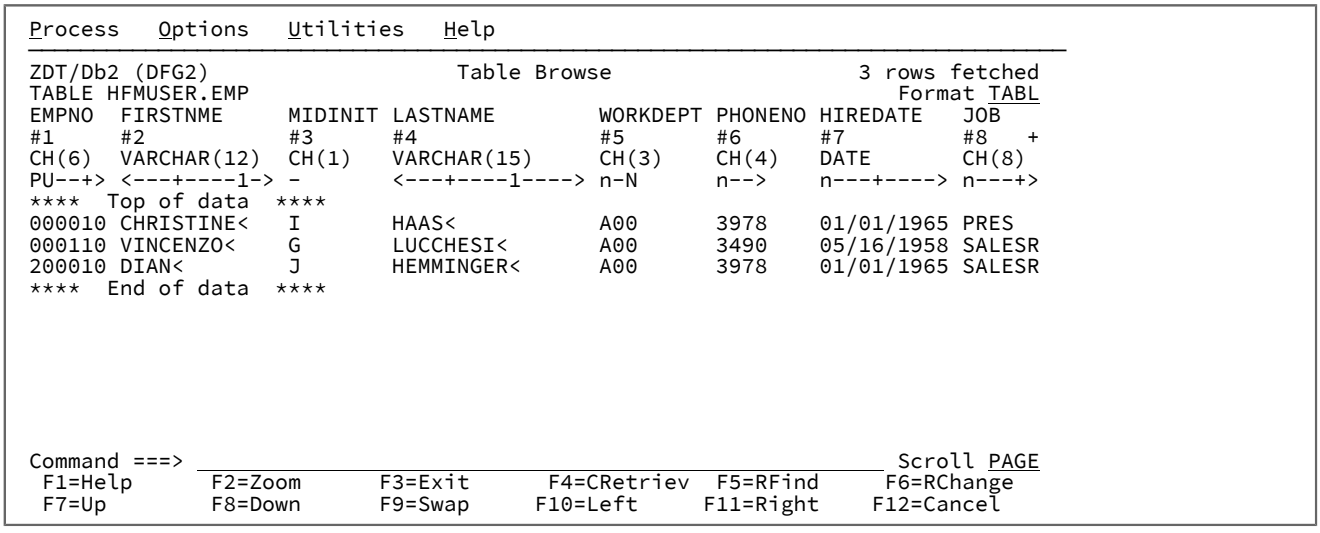


Figure 142. Sample data

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                  Table Browse                  44 rows fetched
TABLE HFMUSER.EMP                                                Format TABL
EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE   JOB
#1     #2           #3      #4              #5       #6      #7         #8   +
CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE       CH(8)
PU--+> <---+----1-> -       <---+----1----> <-NF     <-->    <---+----> <---+>
****  Top of data  ****
000010 CHRISTINE<   I       HAAS<           A00      3978    01/01/1965 PRES
000020 MICHAEL<     L       THOMPSON<       B01      3476    10/10/1973 MANAGE
000030 SALLY<       A       KWAN<           C01      4738    04/05/1975 MANAGE
000050 JOHN<        B       GEYER<          E01      6789    08/17/1949 MANAGE
000060 IRVING<      F       STERN<          D11      6423    09/14/1973 MANAGE
000070 EVA<         D       PULASKI<        D21      7831    09/30/1980 MANAGE
000090 EILEEN<      W       HENDERSON<      E11      5498    08/15/1970 MANAGE
000100 THEODORE<    Q       SPENSER<        E21      0972    06/19/1980 MANAGE
000110 VINCENZO<    G       LUCCHESI<       A00      3490    05/16/1958 SALESR
000120 SEAN<                O'CONNELL<      A00      2167    12/05/1963 CLERK
000130 DOLORES<     M       QUINTANA<       C01      4578    07/28/1971 ANALYS
000140 HEATHER<     A       NICHOLLS<       C01      1793    12/15/1976 ANALYS
000150 BRUCE<               ADAMSON<        D11      4510    02/12/1972 DESIGN
000160 ELIZABETH<   R       PIANKA<         D11      3782    10/11/1977 DESIGN
000170 MASATOSHI<   J       YOSHIMURA<      D11      2890    09/15/1978 DESIGN
000180 MARILYN<     S       SCOUTTEN<       D11      1682    07/07/1973 DESIGN
000190 JAMES<       H       WALKER<         D11      2986    07/26/1974 DESIGN
000200 DAVID<               BROWN<          D11      4501    03/03/1966 DESIGN
000210 WILLIAM<     T       JONES<          D11      0942    04/11/1979 DESIGN
000220 JENNIFER<    K       LUTZ<           D11      0672    08/29/1968 DESIGN
000230 JAMES<       J       JEFFERSON<      D21      2094    11/21/1966 CLERK
000240 SALVATORE<   M       MARINO<         D21      3780    12/05/1979 CLERK
000250 DANIEL<      S       SMITH<          D21      0961    10/30/1969 CLERK
000260 SYBIL<       V       JOHNSON<        D21      8953    09/11/1975 CLERK
000270 MARIA<       L       PEREZ<          D21      9001    09/30/1980 CLERK
000280 ETHEL<       R       SCHNEIDER<      E11      8997    03/24/1967 OPERAT
000290 JOHN<        R       PARKER<         E11      4502    05/30/1980 OPERAT
000300 PHILIP<      X       SMITH<          E11      2095    06/19/1972 OPERAT
000310 MAUDE<       F       SETRIGHT<       E11      3332    09/12/1964 OPERAT
000320 RAMLAL<      V       MEHTA<          E21      9990    07/07/1965 FIELDR
000330 WING<                LEE<            E21      2103    02/23/1976 FIELDR
000340 JASON<       R       GOUNOT<         E21      5698    05/05/1947 FIELDR
200010 DIAN<        J       HEMMINGER<      A00      3978    01/01/1965 SALESR
200120 GREG<                ORLANDO<        A00      2167    05/05/1972 CLERK
200140 KIM<         N       NATZ<           C01      1793    12/15/1976 ANALYS
200170 KIYOSHI<             YAMAMOTO<       D11      2890    09/15/1978 DESIGN
200220 REBA<        K       JOHN<           D11      0672    08/29/1968 DESIGN
200240 ROBERT<      M       MONTEVERDE<     D21      3780    12/05/1979 CLERK
200280 EILEEN<      R       SCHWARTZ<       E11      8997    03/24/1967 OPERAT
200310 MICHELLE<    F       SPRINGER<       E11      3332    09/12/1964 OPERAT
200330 HELENA<              WONG<           E21      2103    02/23/1976 FIELDR
200340 ROY<         R       ALONZO<         E21      5698    05/05/1947 FIELDR
****  End of data  ****

Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Parent panels

- Db2 Browse panel on page 612
- Column Selection/Edit panel on page 497

Child panels

None.

Related tasks

Table Check Constraint Dependencies panel

ZDT/Db2 displays the Table Check Constraint Dependencies panel if you use the DEP line command against an item listed on the Table Check Constraints panel on page 868.

Table Check Constraint Details panel

ZDT/Db2 displays the Table Check Constraint panel if you use the I line command against an item listed on the Table Check Constraints panel on page 868.

Table Check Constraints panel

ZDT/Db2 displays the Table Check Constraints panel if you use the CK line command against an item listed on the Tables, Views and Aliases panel on page 891.

Table Edit panel

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Table Edit                   42 rows fetched
TABLE HFMUSER.EMP                                                Format TABL
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PU--+> <---+----1-> -       <---+----1----> n-N      n-->    n---+---->
000000 ****  Top of data  ****
000001 000010 CHRISTINE<   I       HAAS<           A00      3978    01.01.1965
000002 000020 MICHAEL<     L       THOMPSON<       B01      3476    10.10.1973
000003 000030 SALLY<       A       KWAN<           C01      4738    05.04.1975
000004 000050 JOHN<        B       GEYER<          E01      6789    17.08.1949
000005 000060 IRVING<      F       STERN<          D11      6423    14.09.1973
000006 000070 EVA<         D       PULASKI<        D21      7831    30.09.1980
000007 000090 EILEEN<      W       HENDERSON<      E11      5498    15.08.1970
000008 000100 THEODORE<    Q       SPENSER<        E21      0972    19.06.1980
000009 000110 VINCENZO<    G       LUCCHESI<       A00      3490    16.05.1958
000010 000120 sean<                O'CONNELL<      A00      2167    05.12.1963
000011 000130 DOLORES<     M       QUINTANA<       C01      4578    28.07.1971
000012 000140 HEATHER<     A       NICHOLLS<       C01      1793    15.12.1976
000013 000150 BRUCE<       A       ADAMSON<        D11      4510    12.12.1972
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Prefix area

In TABL format mode, you can enter the following commands in the prefix area of the data display:

A

Identifies the row after which data is to be moved or copied.

B

Identifies the row before which data is to be moved or copied.

C

Copy one row.

C*n*

Copy *n* rows.

CC

Copy block of rows. Mark start and end of block.

D

Delete one row.

D*n*

Delete *n* rows.

DD

Delete block of rows. Mark start and end of block.

E

Display a panel containing information about why a change to this row failed. The E command is only valid when a row is marked in error. Rows marked in error have *"=ERR"*, *"ERR R"*, or the SQLCODE in the prefix field.

F

Display the first row of a block of excluded rows.

F*n*

Display the first *n* rows of a block of excluded rows.

FS

Change the display format to SNGL and display the row.

G

Get the data stored on the clipboard. Rows saved on the clipboard with the S, Sn or SS prefix commands can only be retrieved from the clipboard with the G prefix command during the same Edit session. The clipboard is a temporary memory buffer that is cleared when you exit from the Db2® Edit function panel (and ZDT/Db2 returns you to the Primary Option Menu).

I

Insert one row. The row is initialized using the row description of the currently displayed row. For each column, if it has a default value (DEFAULT="value", and not just the DEFAULT keyword), then it is initialized to that value. Otherwise, the column is initialized according to its data type:

Character strings

Set to blanks

Numeric

Set to zeros

Row identifier

Set to blanks

DATE

Set to the current date

TIME

Set to the current time

TIMESTAMP

Set to the current date and time

Note:
If a column of the inserted row can contain a null value, it is set to the null value and the null column display indicator is displayed in the displayed column.

For more information about setting the value of the null column display indicator, see View and Edit options (options 1 and 2) on page 67.

Related tasks

- View and Edit options (options 1 and 2) on page 67

Related references

- Db2 data types on page 440

I*n*

Insert *n* empty rows.

IP

Insert a primary key. In an Edit session of a related parent table only, insert a row with the key columns set to the key value that originally caused the error in the dependent table of the relationship.

IP*n*

Insert *n* primary keys. In an Edit session of a related parent table only, insert *n* rows with the key columns set to the key value that originally caused the error in the dependent table of the relationship. If you attempt to save the data, it results in duplicate key errors for the second and all subsequent inserted rows, as primary key values must be unique.

J

Join together two or more records - as shown in TABL display mode - into a single, longer record. The record where the prefix command is issued is the record that becomes longer. Any repetition count (for example, *"J4"*) is ignored. The J command can only be issued against a visible record. The join process continues downwards from the current location until:

- The new length for the record would exceed the maximum allowed.
- The last record in the editor session is encountered.

This prefix command is only available in a Z Data Tools/Db2 LOB editor session.

L

Display the last row of a block of excluded rows.

L*n*

Display the last *n* rows of a block of excluded rows.

LC

Convert all uppercase characters in a row to lowercase.

Note: The LC (LC*n*, LCC) command affects *all* characters in a row, not just characters in columns with a CHAR or VARCHAR data type. Therefore, numeric data, such as binary data or packed decimal data, can be corrupted when you use these commands.

LC*n*

Convert all uppercase characters in *n* rows to lowercase. (See note for **LC**.)

LCC

Convert all uppercase characters in a block of rows to lowercase. Mark start and end of block. (See note for **LC**.)

M

Move one row.

M*n*

Move *n* rows.

MM

Move block of rows. Mark start and end of the block.

O

Overlay one row.

O*n*

Overlay *n* rows.

OO

Overlay block of rows. Mark start and end of the block.

P

Identifies the row preceding which data is to be moved or copied .

R or "

Repeat one row.

R*n* or "*n*

Repeat *n* rows.

RE

Edit a related table. You can use the RE prefix command to edit the related table when the row is marked with a error caused by a referential constraint.

You can only enter the RE prefix command against a row marked in error, where the rightmost pos ition of the prefix area contains an *"R"*. RE starts another ZDT/Db2 *dependent* edit session, in a new ISPF logical session. The new (dependent) ZDT/Db2 Edit session is for the related table in the relationship that caused the referential constraint error. For more details, see Editing related tables on page 192.



Note: When you use the RE prefix command for a row whose prefix area contains *"=ERR R"*, take care to avoid the editor interpreting the RE command as an R prefix command instead by ensuring the *"RE"* is followed by one or more blanks).

The RE command lets you quickly identify and resolve referential constraint errors. After you use the RE command to edit a related table, you cannot save data until you end the dependent Edit session of the related table.

This restriction also applies to tables that are part of a larger referential structure. For example, say A, B and C are related tables and A and B are part of a relationship in which A is the parent

table, and B and C are part of a relationship in which B is the parent table. If you edit table A, then use the RE command to edit related table B, then use the RE command in the Edit session of table B to edit table C, it would result in 3 ZDT/Db2 Edit sessions. The Edit sessions for tables B and C are both dependent edit sessions of the Edit session for table A. You cannot save data in the Edit session for table A until you end the Edit sessions for tables B and C.

RR or ""

Repeat block of rows. Mark start and end of the block.

RR*n* or ""*n*

Repeat block of rows *n* times. Mark start and end of block.

S

Save row on clipboard. The clipboard is a temporary memory buffer that is cleared when you exit from the Db2® Edit function panel (and ZDT/Db2 returns you to the Primary Option Menu).

S*n*

Save *n* rows on the clipboard.

SP

Split a single record into multiple pieces, the length of each piece depends on the terminal characteristics. Each piece appears in TABL display mode with the end of the record marker displayed - in other words all of the data for each piece is visible in TABL display mode without the need to scroll.

This prefix command is only available in a Z Data Tools/Db2 LOB editor session.

SS

Save blocks of rows on the clipboard. Mark start and end of block.

UC

Convert all lowercase characters in a row to uppercase.

Note: The UC (UC*n*, UCC) command affects *all* characters in a row, not just characters in columns with a CHAR or VARCHAR data type. Therefore, numeric data, such as binary data or packed decimal data, can be corrupted when you use these commands.

UC*n*

Convert all lowercase characters in *n* rows to uppercase. (See note for **UC**.)

UCC

Convert all lowercase characters in a block of rows to uppercase. Mark start and end of block. (See note for **UC**.)

X

Exclude row from display. If the display of excluded row shadow lines is enabled (see SHADOW primary command on page 1000), a shadow line is shown indicating how many rows are being excluded at this position. To display an excluded row, use the F or LA prefix command.

X*n*

Exclude *n* rows from display. If the display of excluded row shadow lines is enabled (see SHADOW primary command on page 1000), a shadow line is shown indicating how many rows are being excluded at this position.

XX

Exclude block of rows from display. Mark start and end of block. If the display of excluded row shadow lines is enabled (see SHADOW primary command on page 1000), a shadow line is shown indicating how many rows are being excluded at this position.

.xxxx

Label a row. The label, *xxxx*, is a string of 1 to 4 alphabetic characters, that does not start with the character "Z" (labels starting with "Z" are system labels). Labels can be assigned to any row. Labels cannot be assigned to shadow lines.

Parent panels

- Db2 Edit panel on page 616
- Column Selection/Edit panel on page 497

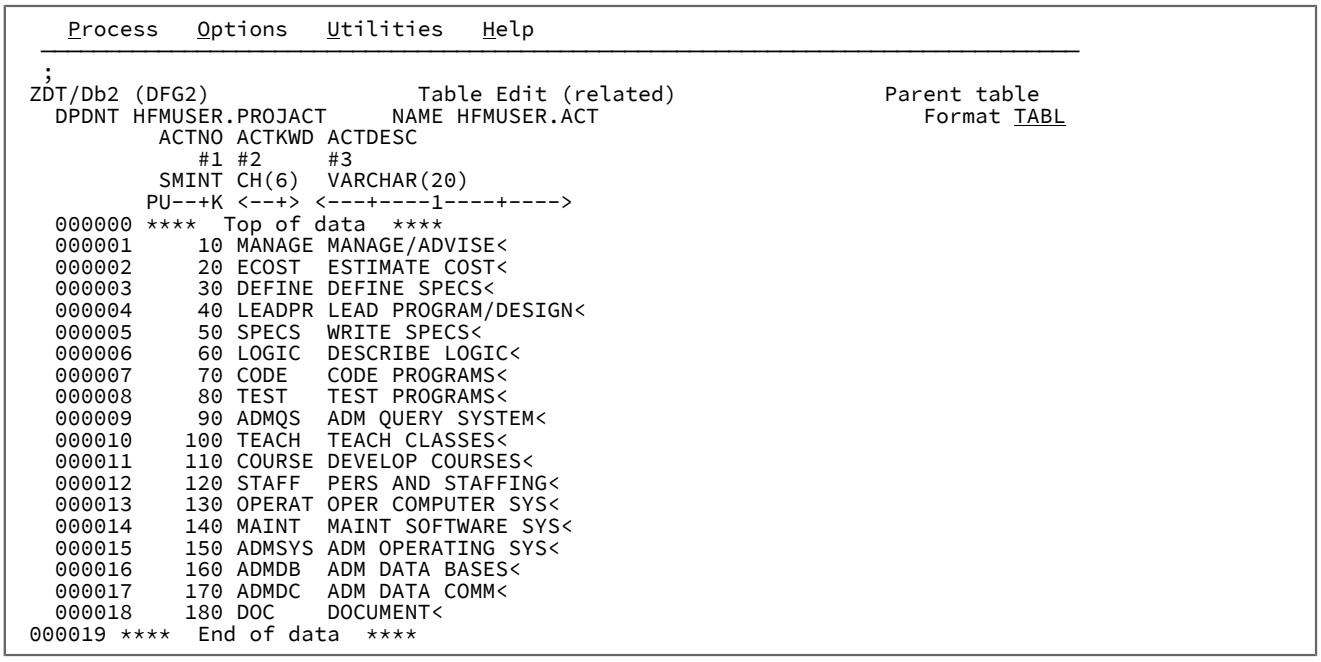
Child panels

None.

Related tasks

Table Edit (related) panel

Panel and field definitions



```
   Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
 ;
ZDT/Db2 (DFG2)                Table Edit (related)               Parent table
  DPDNT HFMUSER.PROJACT     NAME HFMUSER.ACT                      Format TABL
          ACTNO ACTKWD ACTDESC
             #1 #2     #3
           SMINT CH(6)  VARCHAR(20)
         PU--+K <--+> <---+----1----+---->
  000000 ****  Top of data  ****
  000001     10 MANAGE MANAGE/ADVISE<
  000002     20 ECOST  ESTIMATE COST<
  000003     30 DEFINE DEFINE SPECS<
  000004     40 LEADPR LEAD PROGRAM/DESIGN<
  000005     50 SPECS  WRITE SPECS<
  000006     60 LOGIC  DESCRIBE LOGIC<
  000007     70 CODE   CODE PROGRAMS<
  000008     80 TEST   TEST PROGRAMS<
  000009     90 ADMQS  ADM QUERY SYSTEM<
  000010    100 TEACH  TEACH CLASSES<
  000011    110 COURSE DEVELOP COURSES<
  000012    120 STAFF  PERS AND STAFFING<
  000013    130 OPERAT OPER COMPUTER SYS<
  000014    140 MAINT  MAINT SOFTWARE SYS<
  000015    150 ADMSYS ADM OPERATING SYS<
  000016    160 ADMDB  ADM DATA BASES<
  000017    170 ADMDC  ADM DATA COMM<
  000018    180 DOC    DOCUMENT<
000019 ****  End of data  ****
```

Parent panels

This panel appears when you use the REDIT primary command from within an error information panel (providing the error information panel offers REDIT as an option) when a referential integrity error has occurred.

Child panels

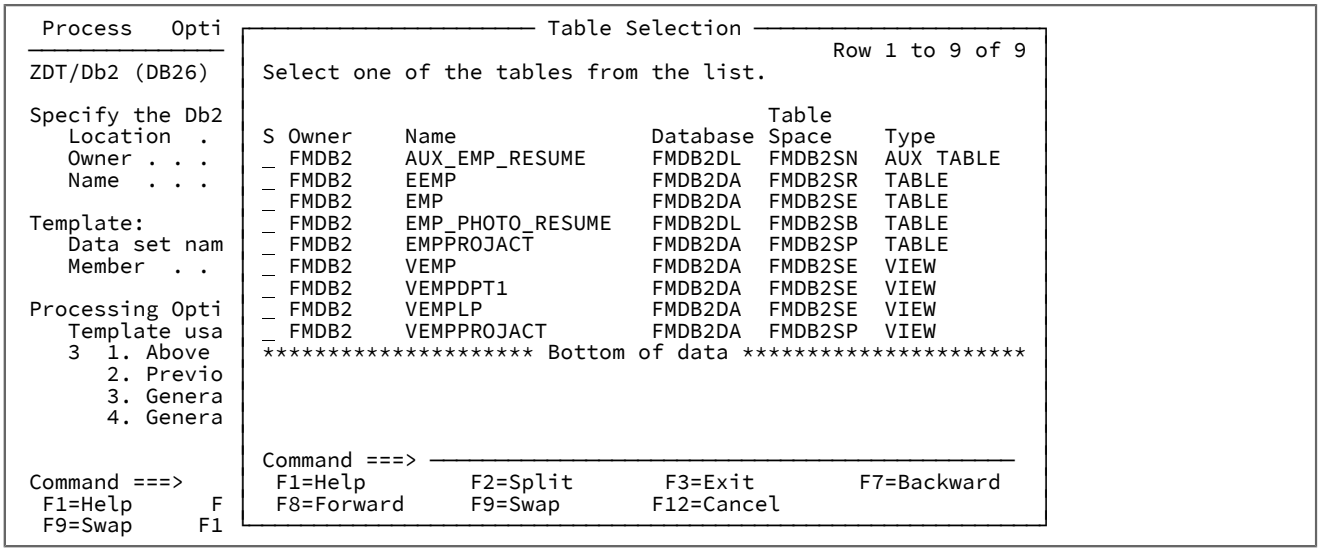
None.

Related tasks

Table Selection panel

You use the Table Selection panel to select a table from a list of tables produced when you have used an asterisk (*) or a pattern when specifying a Db2® object name.

Panel and field definitions



S

Selection column. To select a table, type s.

Owner

The name of the owner of the table.

Name

The name of the table.

Database

The name of the database containing the table.

Table Space

The name of the table space containing the table.

Type

The Db2® object type.

Parent panels

This panel appears when you enter an asterisk (*) or pattern in one or more of the following fields when specifying a Db2® object name: **Owner**, **Name**, **Database**, **Table space**.

Child panels

None.

Related tasks

- Specifying a Db2 object name on page 46

Table Space Parts panel

ZDT/Db2 displays the Table Space Parts panel if you use the PT line command against an item listed on the Table Spaces panel on page 879.

Table Spaces panel

You use the Table Spaces panel to list table space object types in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSTABLESPACE.

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Table Spaces
Location:                                                    Format TABL

SEL    NAME     CREATOR             DBNAME     DBID   OBID   PSID BPOOL
       *        *                   *          *      *      *    *
----   #1--+--- #2--+----1----+--- #3--+--- ----#4 ----#5 ----#6 #7--+---
****  Top of data  ****
_____  DEPTRV8R YEH                 DSNDB04       4     41     42 BP0
_____  DSN32K01 CJUNELL             DSNDB07       7      3      4 BP32K
_____  DSN4K01  CJUNELL             DSNDB07       7      1      2 BP0
_____  DSN8S81C CJUNELL             DSN8D81P    260      1      2 BP0
_____  DSN8S81D CJUNELL             DSN8D81A    259      1      2 BP0
_____  DSN8S81E CJUNELL             DSN8D81A    259      3      4 BP0
_____  DSN8S81P CJUNELL             DSN8D81A    259      7      8 BP0
_____  DSN8S81Q CJUNELL             DSN8D81P    260      3      4 BP0
_____  DSN8S81R CJUNELL             DSN8D81A    259      5      6 BP0
_____  DSN8S81S CJUNELL             DSN8D81A    259      9     10 BP0
_____  DSNASDMA CJUNELL             DSNATPDB    257      5      6 BP0
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Databases panel on page 609
- Tables, Views and Aliases panel on page 891
- Application Packages panel on page 469

Child panels

To display this panel…	Use/do this
Alter Table Space panel on page 469	Line command A
Create Table Space panel on page 564	Line command CR
Create Table panel on page 561	Line command CRT
Databases panel on page 609	Line command D
	Line command DR

To display this panel…	Use/do this
See Drop panels on page 640	
	Line command DS
Database Structure panel on page 609	
	Line command G
Grant privileges panels on page 715	
	Line command GEN
Generate SQL From Db2 Catalog panel on page 712	
	Line command I
Details panels on page 636	
	Line command ICS
Recovery Information panel on page 787	
	Line command P
Privileges panels on page 768	
	Line command PT
Table Space Parts panel on page 879	
	Line command R
Revoke privileges panels on page 798	
	Line command ROW
Display Row panel on page 637	
	Line command T
Tables, Views and Aliases panel on page 891	
	Line command UTL

To display this panel...	Use/do this
Db2 Utilities panel on page 627	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

Table View panel

Panel and field definitions

```
  Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)                    Table View                   42 rows fetched
TABLE HFMUSER.EMP                                                Format TABL
       EMPNO  FIRSTNME     MIDINIT LASTNAME        WORKDEPT PHONENO HIREDATE
       #1     #2           #3      #4              #5       #6      #7
       CH(6)  VARCHAR(12)  CH(1)   VARCHAR(15)     CH(3)    CH(4)   DATE
       PU--+> <---+----1-> -       <---+----1----> n-N      n-->    n---+---->
000000 ****  Top of data  ****
000001 000010 CHRISTINE<   I       HAAS<           A00      3978    01.01.1965
000002 000020 MICHAEL<     L       THOMPSON<       B01      3476    10.10.1973
000003 000030 SALLY<       A       KWAN<           C01      4738    05.04.1975
000004 000050 JOHN<        B       GEYER<          E01      6789    17.08.1949
000005 000060 IRVING<      F       STERN<          D11      6423    14.09.1973
000006 000070 EVA<         D       PULASKI<        D21      7831    30.09.1980
000007 000090 EILEEN<      W       HENDERSON<      E11      5498    15.08.1970
000008 000100 THEODORE<    Q       SPENSER<        E21      0972    19.06.1980
000009 000110 VINCENZO<    G       LUCCHESI<       A00      3490    16.05.1958
000010 000120 sean<                O'CONNELL<      A00      2167    05.12.1963
000011 000130 DOLORES<     M       QUINTANA<       C01      4578    28.07.1971
000012 000140 HEATHER<     A       NICHOLLS<       C01      1793    15.12.1976
000013 000150 BRUCE<       A       ADAMSON<        D11      4510    12.12.1972
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Prefix area

In TABL format mode, you can enter the following commands in the prefix area of the data display:

A

Identifies the row after which data is to be moved or copied.

B

Identifies the row before which data is to be moved or copied.

C

Copy one row.

C*n*

Copy *n* rows.

CC

Copy block of rows. Mark start and end of block.

D

Delete one row.

D*n*

Delete *n* rows.

DD

Delete block of rows. Mark start and end of block.

E

Display a panel containing information about why a change to this row failed. The E command is only valid when a row is marked in error. Rows marked in error have *"=ERR"*, *"ERR R"*, or the SQLCODE in the prefix field.

F

Display the first row of a block of excluded rows.

F*n*

Display the first *n* rows of a block of excluded rows.

FS

Change the display format to SNGL and display the row.

G

Get the data stored on the clipboard. Rows saved on the clipboard with the S, Sn or SS prefix commands can only be retrieved from the clipboard with the G prefix command during the same Edit session. The clipboard is a temporary memory buffer that is cleared when you exit from the Db2® Edit function panel (and ZDT/Db2 returns you to the Primary Option Menu).

I

Insert one row. The row is initialized using the row description of the currently displayed row. For each column, if it has a default value (DEFAULT="value", and not just the DEFAULT keyword), then it is initialized to that value. Otherwise, the column is initialized according to its data type:

Character strings

Set to blanks

Numeric

Set to zeros

Row identifier

Set to blanks

DATE

Set to the current date

TIME

Set to the current time

TIMESTAMP

Set to the current date and time

Note:
If a column of the inserted row can contain a null value, it is set to the null value and the null column display indicator is displayed in the displayed column.

For more information about setting the value of the null column display indicator, see View and Edit options (options 1 and 2) on page 67.

Related tasks

- View and Edit options (options 1 and 2) on page 67

Related references

- Db2 data types on page 440

I*n*

Insert *n* empty rows.

IP

Insert a primary key. In an Edit session of a related parent table only, insert a row with the key columns set to the key value that originally caused the error in the dependent table of the relationship.

IP*n*

Insert *n* primary keys. In an Edit session of a related parent table only, insert *n* rows with the key columns set to the key value that originally caused the error in the dependent table of the relationship. If you attempt to save the data, it results in duplicate key errors for the second and all subsequent inserted rows, as primary key values must be unique.

L

Display the last row of a block of excluded rows.

L*n*

Display the last *n* rows of a block of excluded rows.

LC

Convert all uppercase characters in a row to lowercase.

Note: The LC (LC*n*, LCC) command affects *all* characters in a row, not just characters in columns with a CHAR or VARCHAR data type. Therefore, numeric data, such as binary data or packed decimal data, can be corrupted when you use these commands.

LC*n*

Convert all uppercase characters in *n* rows to lowercase. (See note for **LC**.)

LCC

Convert all uppercase characters in a block of rows to lowercase. Mark start and end of block. (See note for **LC**.)

M

Move one row.

M*n*

Move *n* rows.

MM

Move block of rows. Mark start and end of the block.

P

Identifies the row preceding which data is to be moved or copied .

R or "

Repeat one row.

R*n* or "*n*

Repeat *n* rows.

RE

Edit a related table. You can use the RE prefix command to edit the related table when the row is marked with a error caused by a referential constraint.

You can only enter the RE prefix command against a row marked in error, where the rightmost pos ition of the prefix area contains an *"R"*. RE starts another ZDT/Db2 *dependent* edit session, in a new ISPF logical session. The new (dependent) ZDT/Db2 Edit session is for the related table in the relationship that caused the referential constraint error. For more details, see Editing related tables on page 192.

Note: When you use the RE prefix command for a row whose prefix area contains *"=ERR R"*, take care to avoid the editor interpreting the RE command as an R prefix command instead by ensuring the *"RE"* is followed by one or more blanks).

The RE command lets you quickly identify and resolve referential constraint errors. After you use the RE command to edit a related table, you cannot save data until you end the dependent Edit session of the related table.

This restriction also applies to tables that are part of a larger referential structure. For example, say A, B and C are related tables and A and B are part of a relationship in which A is the parent table, and B and C are part of a relationship in which B is the parent table. If you edit table A, then use the RE command to edit related table B, then use the RE command in the Edit session of table B to edit table C, it would result in 3 ZDT/Db2 Edit sessions. The Edit sessions for tables B and C are both dependent edit sessions of the Edit session for table A. You cannot save data in the Edit session for table A until you end the Edit sessions for tables B and C.

RR or ""

Repeat block of rows. Mark start and end of the block.

RR*n* or ""*n*

Repeat block of rows *n* times. Mark start and end of block.

S

Save row on clipboard. The clipboard is a temporary memory buffer that is cleared when you exit from the Db2® Edit function panel (and ZDT/Db2 returns you to the Primary Option Menu).

S*n*

Save *n* rows on the clipboard.

SS

Save blocks of rows on the clipboard. Mark start and end of block.

UC

Convert all lowercase characters in a row to uppercase.

Note: The UC (UC*n*, UCC) command affects *all* characters in a row, not just characters in columns with a CHAR or VARCHAR data type. Therefore, numeric data, such as binary data or packed decimal data, can be corrupted when you use these commands.

UC*n*

Convert all lowercase characters in *n* rows to uppercase. (See note for **UC**.)

UCC

Convert all lowercase characters in a block of rows to uppercase. Mark start and end of block. (See note for **UC**.)

X

Exclude row from display. If the display of excluded row shadow lines is enabled (see SHADOW primary command on page 1000), a shadow line is shown indicating how many rows are being excluded at this position. To display an excluded row, use the F or LA prefix command.

X*n*

Exclude *n* rows from display. If the display of excluded row shadow lines is enabled (see SHADOW primary command on page 1000), a shadow line is shown indicating how many rows are being excluded at this position.

XX

Exclude block of rows from display. Mark start and end of block. If the display of excluded row shadow lines is enabled (see SHADOW primary command on page 1000), a shadow line is shown indicating how many rows are being excluded at this position.

.xxxx

Label a row. The label, *xxxx*, is a string of 1 to 4 alphabetic characters, that does not start with the character "Z" (labels starting with "Z" are system labels). Labels can be assigned to any row. Labels cannot be assigned to shadow lines.

Parent panels

- Db2 View panel on page 631
- Column Selection/Edit panel on page 497

Child panels

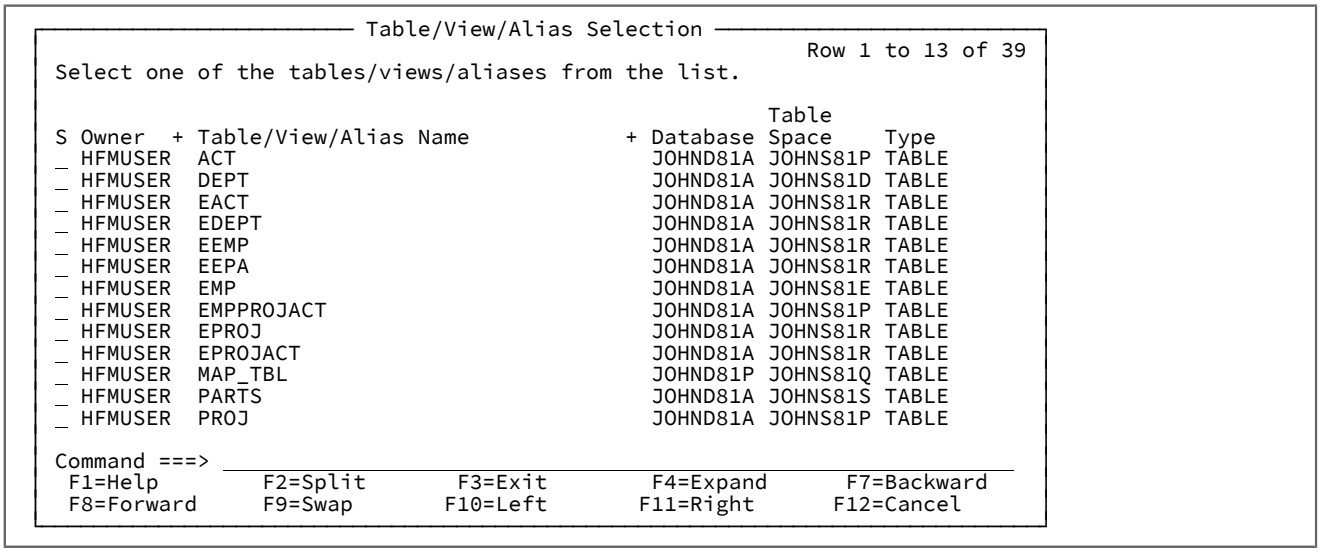
None.

Related tasks

Table/View/Alias Selection panel

You use the Table/View/Alias Selection panel to select a table, view, or alias from a list produced when you have used an asterisk (*) or a pattern when specifying a Db2® object name.

Panel and field definitions



S

Selection column. To select a table, type `s`.

Owner

The name of the owner of the table.

Table/View/Alias Name

The name of the table, view, or alias.

Database

The name of the database containing the table.

Table Space

The name of the table space containing the table.

Type

The Db2® object type.

Parent panels

This panel appears when you enter an asterisk (*) or pattern in one or more of the following fields when specifying a Db2® object name: **Owner**, **Name**, **Database**, **Table space**.

Child panels

None.

Related tasks

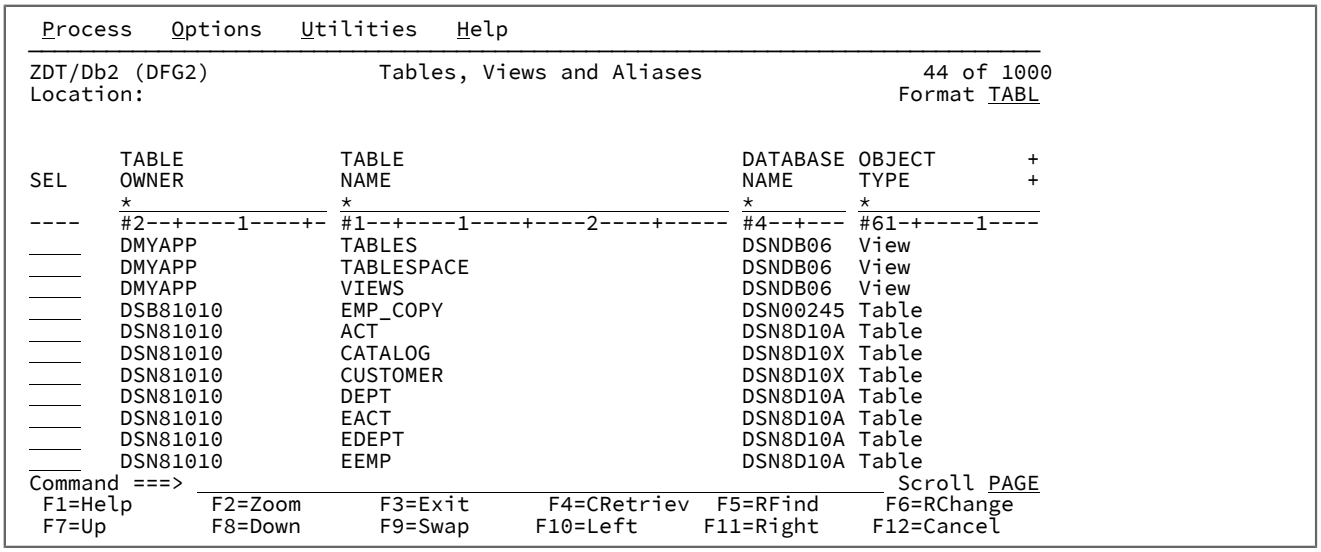
- Specifying a Db2 object name on page 46

Tables, Views and Aliases panel

You use the Tables, Views and Aliases panel to list table, view, and alias object types in the Db2® catalog.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSTABLES.



Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

This panel appears when you specify the T line command against an object displayed in a list of Db2® objects.

Child panels

To display this panel...	Use/do this
	Line command A
Alter Table panel on page 457	

To display this panel…	Use/do this
	Line command ADD
Alter Table - ADD CHECK Constraint panel on page 458	
	Line command ALS
Tables, Views and Aliases panel on page 891	
	Line command AT
Tables, Views and Aliases panel on page 891	
	Line command B
Db2 Browse panel on page 612	
	Line command BT
Tables, Views and Aliases panel on page 891	
	Line command C
Copy Utility ("From") panel on page 513	
	Line command CDI
Column Distribution panel on page 497	
	Line command CH
Tables, Views and Aliases panel on page 891	
	Line command CHR
Children of Table panel on page 482	
	Line command CK
Table Check Constraints panel on page 868	
	Line command COL
Columns in Table panel on page 505	
	Line command COM

To display this panel...	Use/do this
See Comment panels on page 506	
	Line command CR
Create Table panel on page 561	
	Line command CRA
Create Alias panel on page 527	
	Line command CRX
Create Index panel on page 542	
	Line command CS
Columns in Table panel on page 505	
	Line command D
Databases panel on page 609	
	Line command DR
See Drop panels on page 640	
	Line command E
Db2 Edit panel on page 616	
	Line command FK
Foreign Keys for Table panel on page 708	
	Line command G
Grant privileges panels on page 715	
	Line command GEN
Generate SQL From Db2 Catalog panel on page 712	
	Line command I
Details panels on page 636	

To display this panel…	Use/do this
	Line command LAB
Label Table panel on page 728	
	Line command P
Privileges panels on page 768	
	Line command PA
Tables, Views and Aliases panel on page 891	
	Line command PAR
Parents of Table panel on page 753	
	Line command PK
Primary Key for Table panel on page 760	
	Line command PKG
Application Packages panel on page 469	
	Line command PL
Application Plans panel on page 473	
	Line command PR
Print Utility panel on page 764	
	Line command R
Revoke privileges panels on page 798	
	Line command ROW
Display Row panel on page 637	
	Line command S
Table Spaces panel on page 879	
	Line command SEL

To display this panel...	Use/do this
Basic SELECT Prototyping panel on page 476	
	Line command SYN
Synonyms panel on page 855	
	Line command T
Tables, Views and Aliases panel on page 891	
	Line command UTL
Db2 Utilities panel on page 627	
	Line command V
Db2 View panel on page 631	
	Line command VS
Create View Source Statements panel on page 603	
	Line command X
Indexes panel on page 723	
	Line command XC
Indexes and Columns for panel on page 726	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related references

- SORT primary command on page 1003

TEMPLATE Options (1 of 2) panel

You use the TEMPLATE Options (1 of 2) panel to enter information used by ZDT/Db2 to generate TEMPLATE statements and the proper JCL statement to access the TEMPLATE library in a utility batch job.

Panel and field definitions

```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
ZDT/Db2 (DFG2)            TEMPLATE Options (1 of 2)           Global Settings
                                                                  More:     +
Enter RESET to restore installation defaults.

TEMPLATE Library Option:
   JCL DD statement:              No DD stmt is generated if field is blank
      //SYSTEMPL DD ____________________________________________________
      //          DD ____________________________________________________
      //          DD ____________________________________________________
      //          DD ____________________________________________________
Initial Utility Statement Option:
   Utility statement:             No statement is generated if field is blank
      TEMPLATE ____________________________________________________________
      ____________________________________________________________________
      ____________________________________________________________________
      ____________________________________________________________________
   Additional statements:
      Enter "/" to select options
      _  Use additional statements
      _  Edit additional statements
Command ===> ________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap     F11=NxtPage  F12=Cancel
```

JCL DD statement

The information used to generate the JCL DD statement. This information is:

DD name

A read-only field. This holds the value that was specified in the **TEMPLATEDD DD name** field on the OPTIONS Options panel (OPTIONS Options panel on page 750). If the library field is blank then no DD statement is generated in the utility batch job.

Text values

The values entered on these four lines are used directly to generate the DD statement, provided that the **DD name** field is not blank. The default is blank.

Utility statement

The field of the Utility statement is used to generate the initial utility statements. If the field is blank, and the **Use additional statements** option is not selected, then no TEMPLATE statement is generated into the utility batch job. If the input field is non-blank, then the specified TEMPLATE is generated with any additional TEMPLATE statements supplied when the **Use additional statements** option is selected. The default is blank.

Use additional statements

If this option is set then more than one TEMPLATE statement may be generated into the utility batch job. Select the Edit additional statements option to edit additional statements. Only TEMPLATE specifications that have non-blank input fields are used in the generation. The default is not selected.

If this option is not selected, then only a non-blank value in the **Utility statement** field is used to generate the TEMPLATE statement.

Edit additional statements

When you select this option and press Enter, the Additional TEMPLATE Statements panel (Additional TEMPLATE Statements panel on page 450) is displayed. After you have entered values on this panel and exited the panel, this option is deselected. The additional statements are used when you select the Use additional statements option. The default is not selected.

The TEMPLATE Library Option panel has an installation-defined default. You can restore the default for the library option by entering the RESET command or pressing the RESET PF key. Each subsystem has its own installation-defined defaults. RESET also clears the **Use additional statements** option and any values in the **Utility statement** fields.

For more information, see TEMPLATE and OPTIONS statements in *DB2 Utility Guide and Reference*.

Child panels

To display this panel…	Use/do this
Additional LISTDEF Statements panel on page 449	Select the Edit additional statements option.
TEMPLATE Options (2 of 2) panel on page 898	Press F11

Related tasks

- RESET primary command on page 994
- Setting options for Db2 utilities on page 348
- Db2 Utility TEMPLATE options (option 3.UT) on page 70

TEMPLATE Options (2 of 2) panel

You use the TEMPLATE Options (2 of 2) panel to enter more information used by ZDT/Db2 to generate TEMPLATE statements and the proper JCL statement to access the TEMPLATE library in a utility batch job.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)           TEMPLATE Options (2 of 2)           Global Settings

Enter RESET to restore installation defaults.

Utility DD-Keyword Option:
  COPY Statement:
     COPYDDN template name . _________
  LOAD Statement:
     INDDN template name . . _________
  REBUILD INDEX Statement:
     WORKDDN template name . _________
  REORG TABLESPACE Statement:
     UNLDDN template name  . _________
  UNLOAD Statement:
     PUNCHDDN template name  _________
     UNLDDN template name  . _________


Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F6=Reset     F7=Backward  F8=Forward
 F9=Swap     F10=PrvPage  F12=Cancel
```

COPYDDN template name

A new ZDT/Db2 default setting for a COPYDDN clause. The original default is the DD-name SYSCOPY. The default value for this field is blank.

INDDN template name

A new ZDT/Db2 default setting for a INDDN clause. The original default is the DD-name SYSREC00. The default value for this field is blank.

WORKDDN template name

A new ZDT/Db2 default setting for a WORKDDN clause. The original default is SYSUT1. The default value for this field is blank.

UNLDDN template name

A new ZDT/Db2 default setting for a UNLDDN clause. The default value for this field is blank.

The UNLDDN clause is only generated if the Unload External option of the REORG Utility (Table Spaces) panel (REORG Utility (Table Spaces) panel on page 795) is set to *"Y"*.

The original default is either the DD-name SYSREC00 or SYSREC. It depends on a number of factors including what options are chosen.

PUNCHDDN template name

A new ZDT/Db2 default setting for the PUNCHDDN clause. The original default is the DD-name SYSPUNCH. The default value for this field is blank.

UNLDDN template name

A new ZDT/Db2 default setting for the UNLDDN clause. The original default is the DD-name SYREC00. The default value for this field is blank.

Parent panels

- TEMPLATE Options (1 of 2) panel on page 897

Child panels

None.

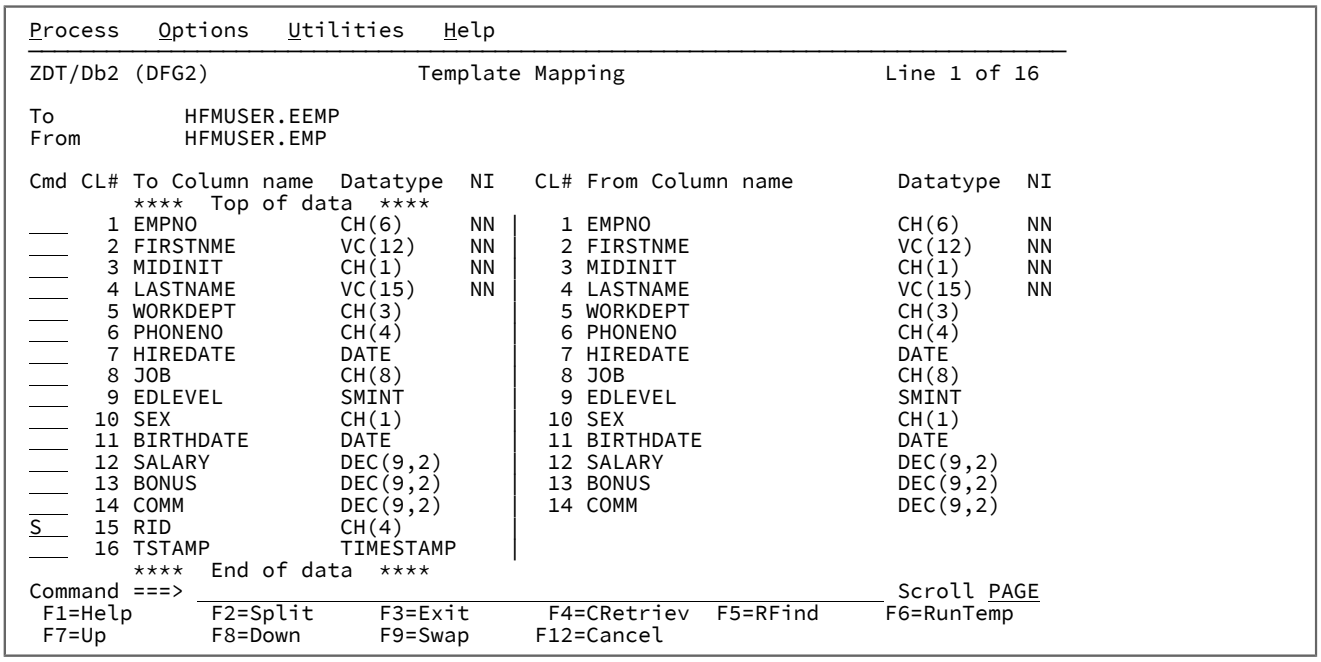
Related tasks

- RESET primary command on page 994
- Setting options for Db2 utilities on page 348
- COPY utility on page 354
- LOAD utility on page 357
- Loading data described by a copybook or template on page 361
- REBUILD utility (indexes) on page 366
- REORG (table spaces) on page 372
- UNLOAD utility on page 380
- UNLOAD (tables) on page 381
- UNLOAD (table spaces) on page 386
- Db2 Utility TEMPLATE options (option 3.UT) on page 70

Template Mapping panel

You use the Template Mapping panel to specify the columns in the source (the *"From"* columns) you want transferred to the columns in the target (the *"To"* columns) when you are moving (copying) data.

Panel and field definitions



TO

The *"To"* Db2® object. The Db2® object to which data will be imported.

FROM

The *"From"* data set. The name of the import data set.

Cmd

Prefix command area, used to enter a template editor prefix command. S or E to edit the column mapping.

Cl# (TO)

The *"To"* column number. The number of the column in the *"To"* template.

Column name (TO)

The *"To"* column name. The name of the column in the *"To"* template.

Datatype (TO)

The *"To"* data type. The data type and, optionally, the length of the column in the *"To"* template.

NI (TO)

The *"To"* Null Indicator. The null indicator for the column in the *"To"* template:

(blank)

The column allows null values

NN

The column does not allow null values.

Cl# (FROM)

The *"From"* column number. The number of the column in the *"From"* template.

Column name (FROM)

The *"From"* column name. The name of the column in the *"From"* template.

Datatype (FROM)

The *"From"* data type. The data type and, optionally, the length of the column in the *"From"* template.

NI (FROM)

The *"From"* Null Indicator. The null indicator for the column in the *"From"* template:

(blank)

The column allows null values

NN

The column does not allow null values.

Parent panels

- Copy Utility ("To") panel on page 521
- Import Utility ("To") panel on page 720
- Export Utility ("To") panel on page 701
- LOAD from panel on page 731

Child panels

To display this panel…	Use/do this
From Column Mapping panel on page 709	Enter *"S"* in **Cmd** field

Related tasks

- Specifying your own mapping on page 114
- Copying data from one Db2 object to another on page 246
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279
- LOAD utility on page 357

Template Save panel

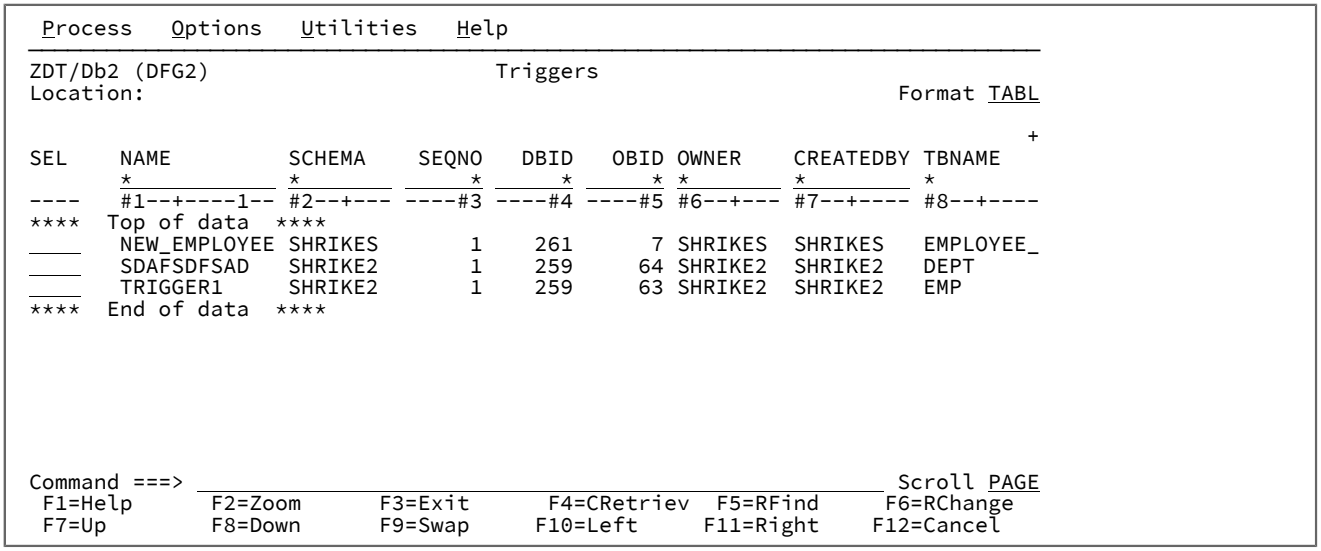
ZDT/Db2 displays the Template Save panel if you press F3 when the Column Selection/Edit panel on page 497 is displayed.

Triggers panel

You use the Triggers panel to display information about the triggers you have selected.

Panel and field definitions

The columns that are displayed include the SEL field and columns of SYSIBM.SYSTRIGGERS.



Note:

1. The system option, **Show all catalog table columns**, affects which columns ZDT/Db2 displays when you use the Object List utility.
2. You can also customize which columns ZDT/Db2 displays when you use the Object List utility. For details, see the *Z Data Tools Customization Guide*.

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Object List Utility panel on page 748
- Schemas panel on page 812

Child panels

To display this panel…	Use/do this
See Comment panels on page 506	Line command COM
Create Trigger panel on page 596	Line command CR
See Drop panels on page 640	Line command DR
Details panels on page 636	Line command I
Application Packages panel on page 469	Line command PKG
Privileges panels on page 768	Line command PS
Display Row panel on page 637	Line command ROW
Schemas panel on page 812	Line command SCH
(ISPF panel)	Line command SQL
	Line command T

To display this panel…	Use/do this
Tables, Views and Aliases panel on page 891	
	Primary command SORT
Sort Fields panel on page 841	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

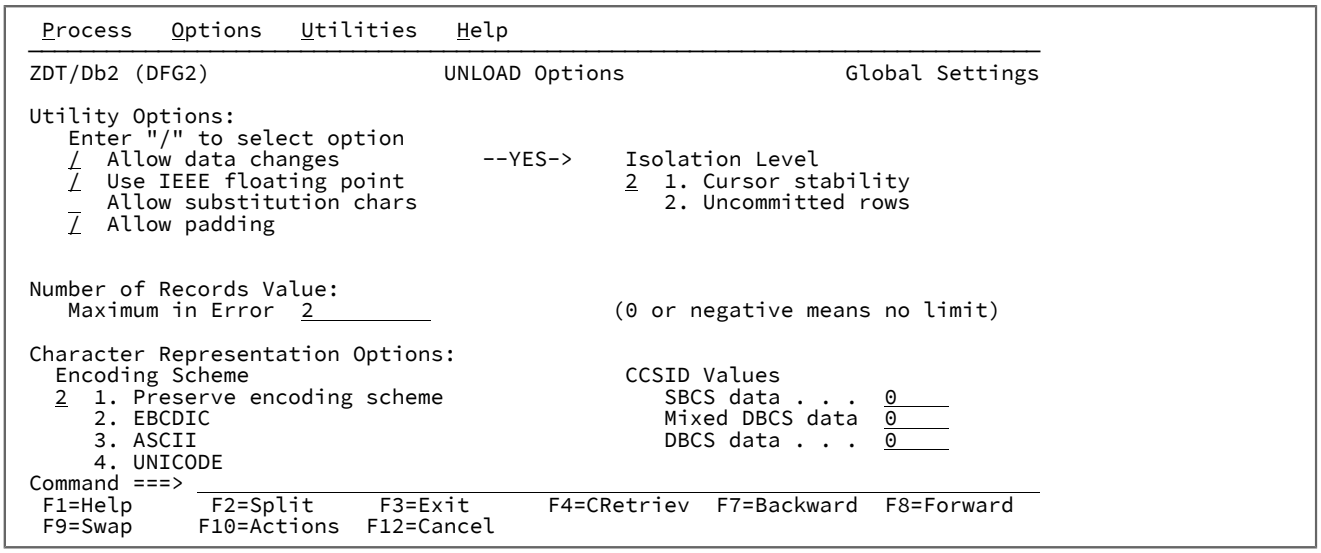
Related references

- SORT primary command on page 1003

UNLOAD Options panel

You use the UNLOAD Options panel to enter information that is used in generating an UNLOAD statement in a utility batch job.

Panel and field definitions



Allow data changes

When this option is selected, you must supply a value for the Isolation Level. The default is selected.

If not selected, ZDT/Db2 adds the SHRLEVEL REFERENCE clause to the UNLOAD utility control statements, meaning that during the unload operation, rows of the tables can be read, but can not be inserted, updated, nor deleted by other Db2® threads.

Isolation Level

This parameter comes into effect when the Allow data changes parameter is selected. The default is 1 (**Cursor stability**). Valid values are:

1 (Cursor stability)

ZDT/Db2 omits a SHRLEVEL clause to the UNLOAD utility statement. This is equivalent to adding a SHRLEVEL CHANGE ISOLATION CS clause to the UNLOAD utility control statement, meaning that rows can be read, inserted, updated, and deleted from the table space or partition while the data is being unloaded and that the UNLOAD utility reads rows in cursor stability mode.

2 (Uncommitted rows)

ZDT/Db2 adds a SHRLEVEL CHANGE ISOLATION UR clause to the UNLOAD utility control statement. This means that rows can be read, inserted, updated, and deleted from the table space or partition while the data is being unloaded and that uncommitted rows, if they exist, are unloaded.

Use IEEE floating point

When this option is selected, ZDT/Db2 adds the FLOAT IEEE clause to the UNLOAD utility control statements.

If not selected, no clause is added; this means that the Db2® UNLOAD utility output format of the numeric floating point numbers are produced in S/390® hexadecimal Floating Point (HFP) format. The default is not selected.

Allow substitution chars

When this option is not selected, ZDT/Db2 adds the NOSUBS clause to the UNLOAD utility control statement. This means that substitution characters are not allowed in strings. The substitution character that is attempted while unloading data is treated as a conversion error.

The default is selected. A substitution character is sometimes placed in a string when that string is being converted from ASCII to EBCDIC, or converted from one CCSID to another. For example, substitution occurs when a character (sometimes referred to as a code point) that exists in the source CCSID (code page) does not exist in the target CCSID (code page).

Allow padding

When this option is selected, the NOPAD clause is not added to the UNLOAD utility control statements. The default is selected.

If not selected, ZDT/Db2 adds the NOPAD clause to the UNLOAD utility control statements.

The Db2® default for UNLOAD processing is to pad variable length columns in the unloaded records to their maximum length, and the unloaded records have the same length for each table. Select the item if you wish to use the Db2® defined default behavior.

When padding is performed:

- The padded data fields are preceded by the length fields that indicate the size of the actual data without the padding.
- When the output records are reloaded using the LOAD utility, padded data fields are treated as varying length data.

Maximum in Error

This option specifies the value for the maximum number of records in error that are allowed. The unloading process terminates when this value is reached. ZDT/Db2 adds a MAXERR clause with specified value to the UNLOAD utility statements. The default value is 1.

Encoding Scheme

This option specifies the encoding scheme used by the generated UNLOAD utility control statement for all output data of the character type. Valid values are:

1 (Preserve encoding scheme)

The encoding scheme of the source data is preserved (default).

2

EBCDIC is the encoding scheme for all output data of the character type.

3

ASCII is the encoding scheme for all output data of the character type.

4

UNICODE is the encoding scheme for all output data of the character type.

CCSID Values

This option specifies the values for the coded character set identifiers (CCSIDs) for the data in the output data sets. If you specify one or more non-zero values for the **SBCS data**, **Mixed DBCS data** and **DBCS data** fields, ZDT/Db2 adds a CCSID clause to the UNLOAD utility control statements. If all are zero, no CCSID clause is added.

SBCS data

Specifies the CCSID value for single byte character set (SBCS) data in output records, including data unloaded in the external character formats.

Mixed DBCS data

Specifies the CCSID value for mixed double byte character set (DBCS) data in output records, including data unloaded in the external character formats.

DBCS data

Specifies the CCSID value for DBCS data in output records, including data unloaded in the external character formats.

Zero is the default value for each of the above three CCSID value fields.

With the example settings shown in the figure, the following clauses are added to any generated UNLOAD:

- SHRLEVEL CHANGE ISOLATION UR
- FLOAT IEEE
- NOSUB
- MAXERR 2
- EBCDIC

Child panels

None.

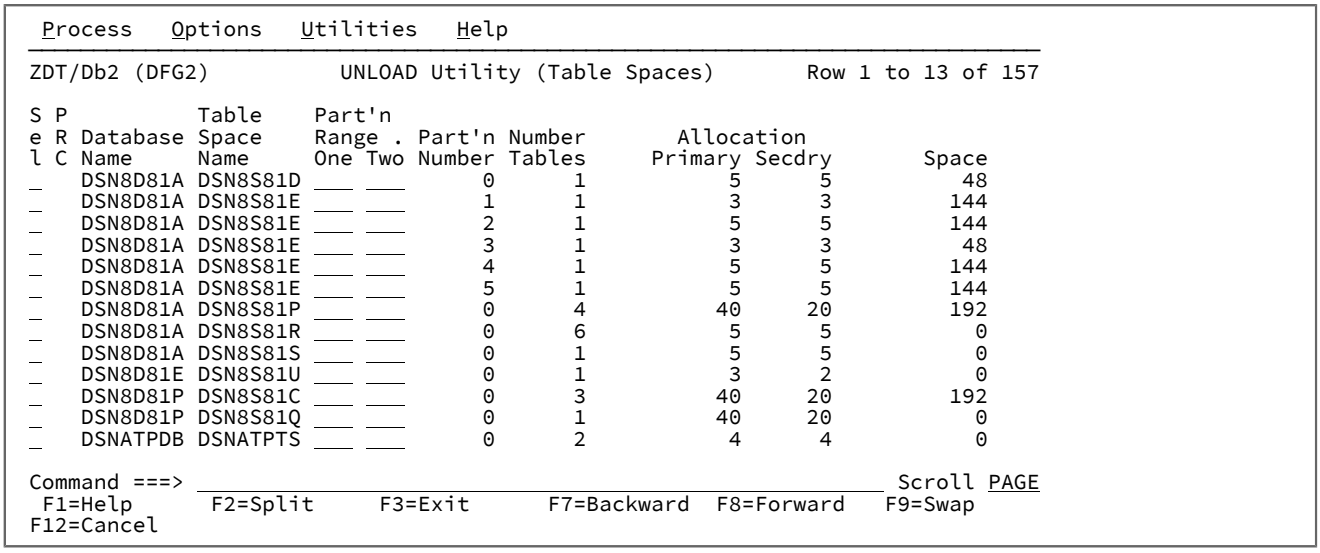
Related tasks

- Setting options for Db2 utilities on page 348
- UNLOAD (tables) on page 381
- Null indicator byte processing on page 385
- UNLOAD (table spaces) on page 386
- Db2 Unload utility options (option 3.UU) on page 70

UNLOAD Utility (Table Spaces) panel

You use the UNLOAD Utility (Table Spaces) panel to generate utility control statements using the UNLOAD TABLESPACE form of the UNLOAD utility.

Panel and field definitions



```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)          UNLOAD Utility (Table Spaces)       Row 1 to 13 of 157

S P          Table    Part'n
e R Database Space    Range . Part'n Number       Allocation
l C Name     Name     One Two Number Tables     Primary Secdry       Space
_   DSN8D81A DSN8S81D ___ ___      0      1           5      5          48
_   DSN8D81A DSN8S81E ___ ___      1      1           3      3         144
_   DSN8D81A DSN8S81E ___ ___      2      1           5      5         144
_   DSN8D81A DSN8S81E ___ ___      3      1           3      3          48
_   DSN8D81A DSN8S81E ___ ___      4      1           5      5         144
_   DSN8D81A DSN8S81E ___ ___      5      1           5      5         144
_   DSN8D81A DSN8S81P ___ ___      0      4          40     20         192
_   DSN8D81A DSN8S81R ___ ___      0      6           5      5           0
_   DSN8D81A DSN8S81S ___ ___      0      1           5      5           0
_   DSN8D81E DSN8S81U ___ ___      0      1           3      2           0
_   DSN8D81P DSN8S81C ___ ___      0      3          40     20         192
_   DSN8D81P DSN8S81Q ___ ___      0      1          40     20           0
_   DSNATPDB DSNATPTS ___ ___      0      2           4      4           0

Command ===> _____________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

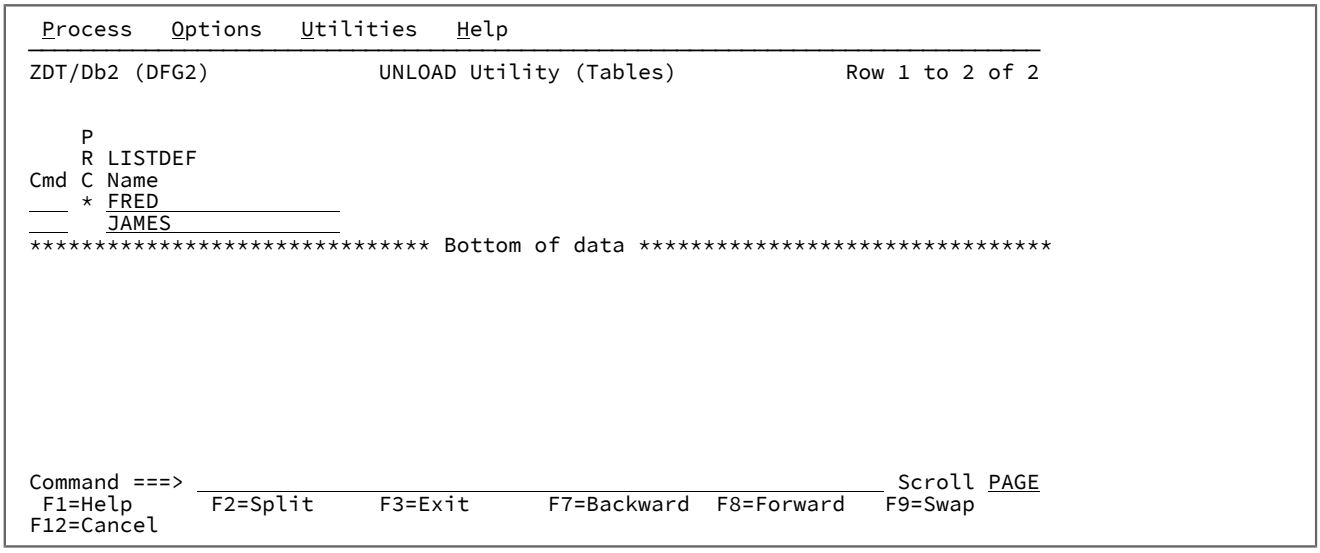
Related tasks

- UNLOAD utility on page 380
- UNLOAD (table spaces) on page 386
- Where you can use templates on page 22

UNLOAD Utility (Table Spaces) with LISTDEF panel

You use the UNLOAD Utility (Table Spaces) with LISTDEF panel to generate utility control statements using the UNLOAD TABLESPACE form of the UNLOAD utility.

Panel and field definitions



Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

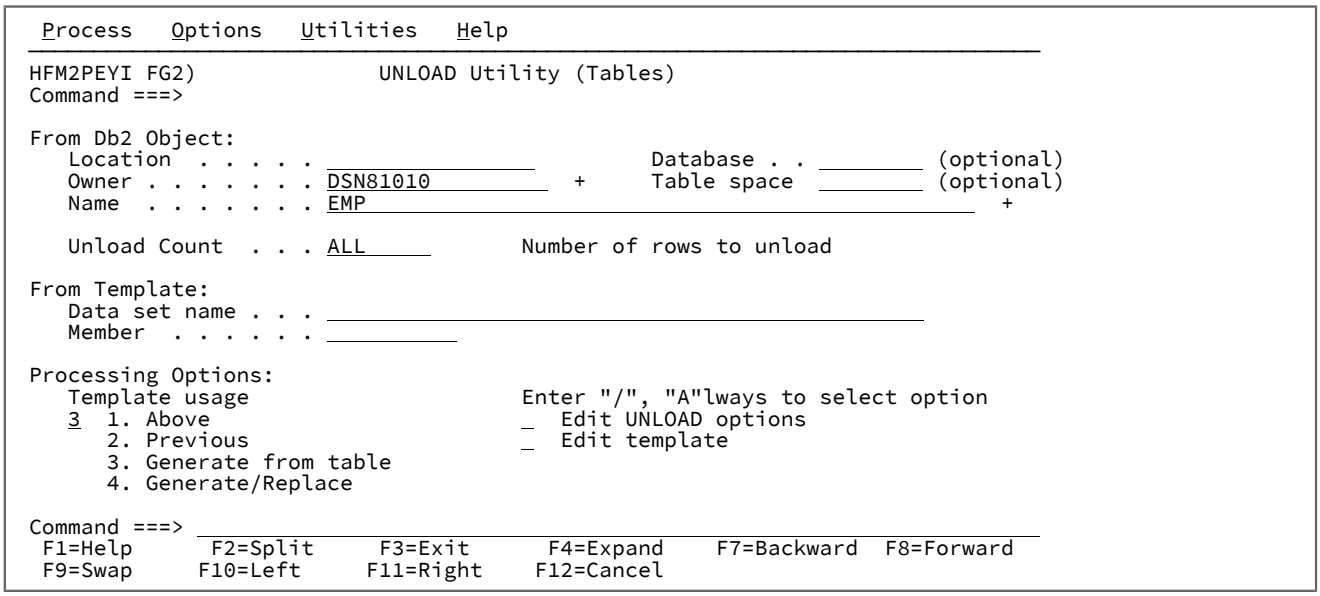
Related tasks

- UNLOAD utility on page 380
- UNLOAD (table spaces) with LISTDEF specified on page 388
- Where you can use templates on page 22

UNLOAD Utility (Tables) "From" panel

You use the UNLOAD Utility (Tables) "From" panel to specify the name of the Db2® object that contains the data to be unloaded, the (optional) name of a template data set, and processing options.

Panel and field definitions



Edit UNLOAD options

Select this option to edit the UNLOAD options. The default is not selected.

Edit template

Select this option to edit the template. The default is not selected.

Parent panels

- Db2 Utilities panel on page 627

Child panels

To display this panel...	Use/do this
UNLOAD Utility (Tables) "To" panel on page 911	Press Enter

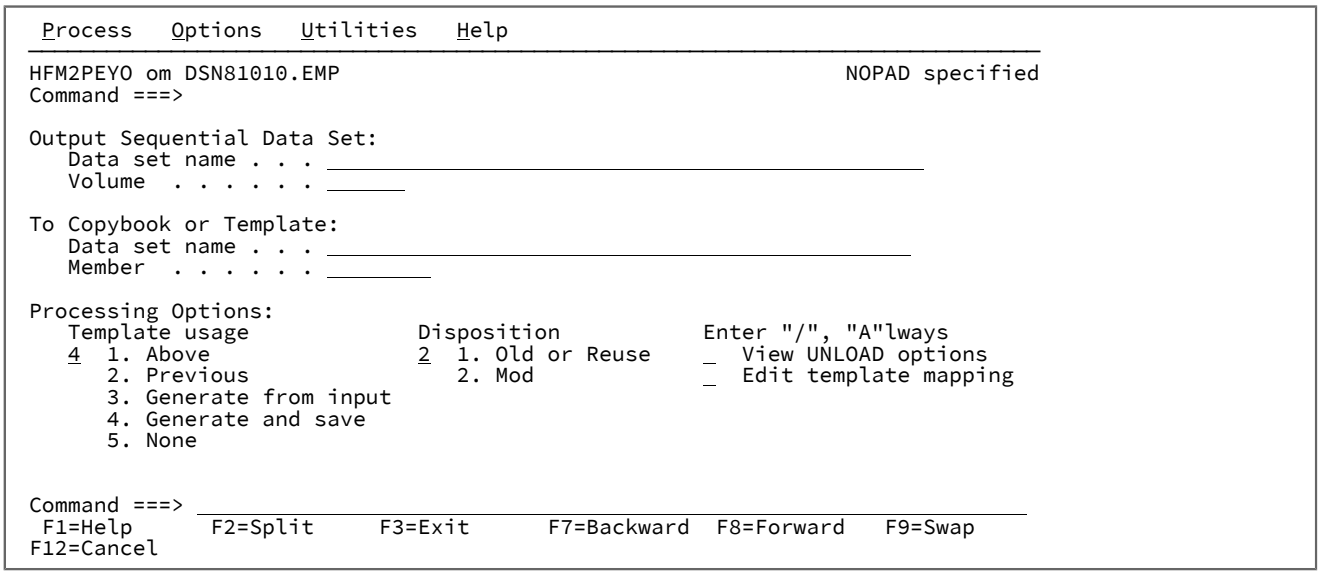
Related tasks

- UNLOAD utility on page 380
- UNLOAD (tables) on page 381
- Where you can use templates on page 22

UNLOAD Utility (Tables) "To" panel

You use the UNLOAD Utility (Tables) "To" panel to specify the name of the data set that will contain the data to be unloaded, the (optional) name of a template data set, and processing options.

Panel and field definitions



Data set name

The name of the data set or PDS member that contains the unloaded data. The name can be a fully qualified data set name or a pattern. The name cannot include a member name, which is a restriction that is imposed by the Db2® UNLOAD utility.

Volume

The volume serial number if the data set is not cataloged.

Data set name (To Copybook or Template)

The name of the sequential or partitioned data set that contains the copybook or template that describes the data set. It can be a fully qualified data set name or a pattern. The name might include a member name or name pattern in parentheses. If the member is specified, the associated **Member** field must be empty.

Member (To Copybook or Template)

If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parentheses in the **Data set name** field, you can use this field to specify the member name or a member name pattern.

Template usage

The **Unload Utility "To"** panel provides five options for template processing:

1. **Above**

   You must enter the name of a copybook or template in the **To Copybook or Template** section of the panel. If you do not specify a member name or if you specify a pattern, ZDT/Db2 displays a member selection list. After you specified the member that contains a copybook or template, processing continues by using this template instead of the template that is automatically generated for the table (see Template usage option 3).

2. Previous

Uses the last template or the template that was previously used for this table.

3. Generate from input

Generates a template based on the input template. However, the template includes the columns that are selected in the input template only. If you specified a data set name in the **To Copybook or Template** section of the panel, this option is ignored. The template that is generated is not saved.

4. Generate and save

Generates a template based on the input template. However, the template includes the columns that are selected in the input template only. To enable this option, you must enter the name of a template data set and optionally a member name in the **To Copybook or Template** section of the panel. If you do not specify a member name, or if you specify a pattern, ZDT/Db2 displays a member selection list. The generated template is saved.

5. None.

ZDT/Db2 does not use an output template.

Disposition

The disposition for the data set that contains the unloaded data.

1. Old or Reuse

Specify '1' when the unload data set exists, and the unloaded data is to replace any existing data.

2. Mod

Specify '2' in one of the following conditions. This is the default.

- The unload data does not exist, and it is to be created.
- The unload data set exists, and the unloaded data is to be added to any existing data.

View UNLOAD options

Select this option to show the view of the current UNLOAD options. The default is not selected.

Edit template mapping

Use to edit the template for the partitioned, sequential, or VSAM data set to which you are unloading the data, before you unload the data. The default is not selected.

Parent panels

- UNLOAD Utility (Tables) "From" panel on page 910

Child panels

None.

Related tasks

- UNLOAD utility on page 380
- UNLOAD (tables) on page 381
- Specifying a data set and a member name on page 52
- Letting ZDT/Db2 do the mapping on page 113
- Specifying your own mapping on page 114

UNLOAD Utility (Tables) with LISTDEF panel

You use the UNLOAD Utility (Tables) with LISTDEF panel to generate utility control statements using the UNLOAD TABLESPACE form of the UNLOAD utility.

Panel and field definitions

```
 Process   Options   Utilities   Help
_____________________________________________________________________________
ZDT/Db2 (DFG2)             UNLOAD Utility (Tables)             Row 1 to 1 of 1


    P
    R LISTDEF
Cmd C Name
___   FRED_____________
****************************** Bottom of data ********************************









Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Cmd

You can enter commands in this field to manipulate the panel rows.

I(n)

Insert (n) new row(s)

R(n)

Repeat the current row (n times)

D(n)

Delete (n) row(s)

S(n)

Select (n) rows(s)

An asterisk (*) in the PRC column shows that the LISTDEF in the row is selected. The S command toggles the selection of rows; unselected rows become selected and selected rows become unselected.

LISTDEF Name

On entry to the panel, this field is set to the value in the **LISTDEF name** field in the Db2® Utilities panel.

Parent panels

- Db2 Utilities panel on page 627

Child panels

None.

Related tasks

- UNLOAD utility on page 380
- UNLOAD (tables) with LISTDEF specified on page 386
- Where you can use templates on page 22

Utility Functions panel

You use the Utility Functions panel to select the ZDT/Db2 utility you want to use.

Panel and field definitions

```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
ZDT/Db2 (DFG2)               Utility Functions

1  Print         Print Db2 table or view
2  Objects       Create and drop Db2 objects
3  Copy          Copy data within Db2
4  Object List   Display and process Db2 object lists
5  Privileges    Manage Db2 privileges
6  Import        Import sequential or VSAM data into Db2
7  Export        Export Db2 data to sequential or VSAM data set
8  Create        Create Db2 test data
9  Utilities     Db2 utility job generation
10 Audit trail   Print audit trail report
11 Print browse  Browse ZDT/Db2 print data set




Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

1 Print

Print Db2® table or view

2 Objects

Create and drop Db2® objects

3 Copy

Copy data within Db2®

4 Object List

Display and process Db2® object lists

5 Privileges

Manage Db2® privileges

6 Import

Import sequential or VSAM data into Db2®

7 Export

Export Db2® data to a sequential or VSAM data set

8 Create

Create Db2® test data

9 Utilities

Db2® utility job generation

10 Audit trail

Print an audit trail report

11 Print browse

Browse your print output

Parent panels

- Primary Option Menu panel on page 760

Child panels

To display this panel…	Use/do this
Print Utility panel on page 764	Option 1
Db2 Object Functions panel on page 622	Option 2
Copy Utility ("From") panel on page 513	Option 3
Object List Utility panel on page 748	Option 4
Manage Db2 Privileges panel on page 741	Option 5
Import Utility ("From") panel on page 719	Option 6

To display this panel…	Use/do this
Export Utility ("From") panel on page 698	Option 7
Data Create Utility panel on page 603	Option 8
Db2 Utilities panel on page 627	Option 9
Print Audit Trail panel on page 761	Option 10

Related tasks

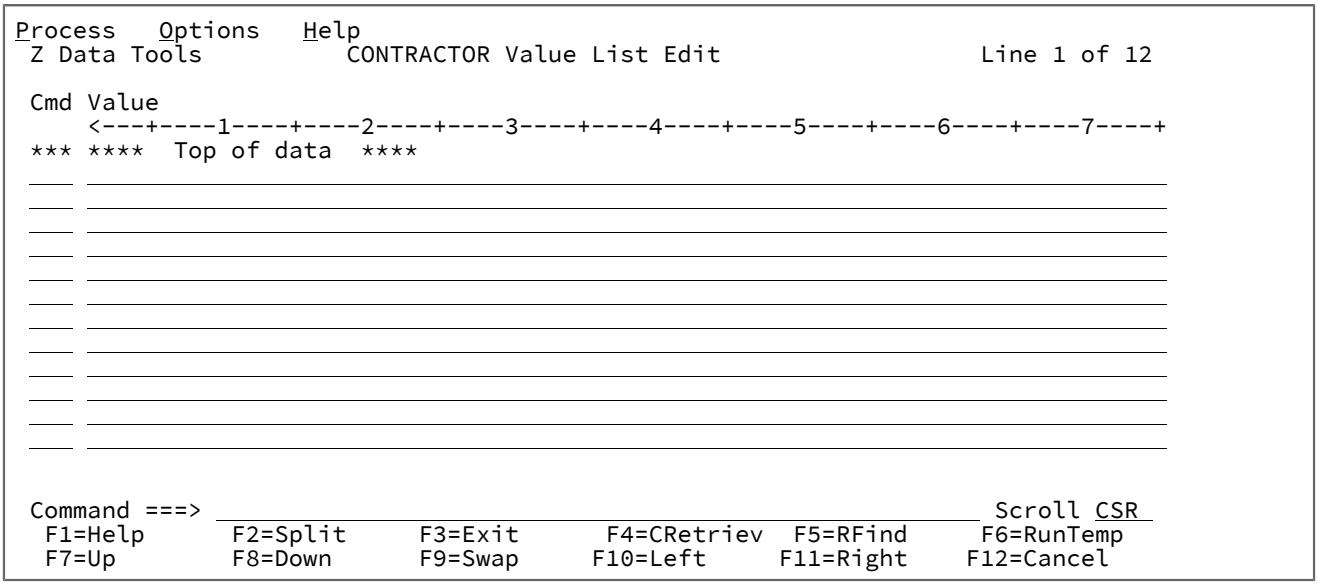
- Printing the contents of a Db2 table on page 328
- Creating and dropping Db2 objects on page 208
- Copying data from one Db2 object to another on page 246
- Working with lists of Db2 objects on page 303
- Managing Db2 privileges on page 339
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279
- Populating a Db2 table with data on page 244
- Generating batch JCL for Db2 utility jobs on page 346
- Printing an audit trail report on page 335
- Looking at the print output from your ZDT/Db2 session on page 333
- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Value List Edit panel

This panel allows you to to create and edit value lists to be used in field scrambling.

Panel and field definitions

Figure 143. Value List Edit panel



Cmd

Prefix command field. Allows you to copy, insert, move, repeat or delete lines. Available commands are:

A

Identifies the line after which data is to be moved or copied.

B

Identifies the line before which data is to be moved or copied.

C

Copy one line.

C*n*

Copy *n* lines.

CC

Copy block of lines. Marks the start and end of the block.

D

Delete one line.

D*n*

Delete *n* lines.

DD

Delete block of lines. Marks the start and end of the block.

I

Insert one empty line.

I*n*

Insert *n* empty lines.

M

Move one line.

M*n*

Move *n* lines.

MM

Move block of lines. Marks the start and end of the block.

R

Repeat line once.

R*n*

Repeat line n times.

RR

Repeat block of lines. Marks the start and end of the block.

RR*n*

Repeat block of lines *n* times. Marks the start and end of the block.

Value

A value that can be used when field scrambling is in effect.

Parent panels

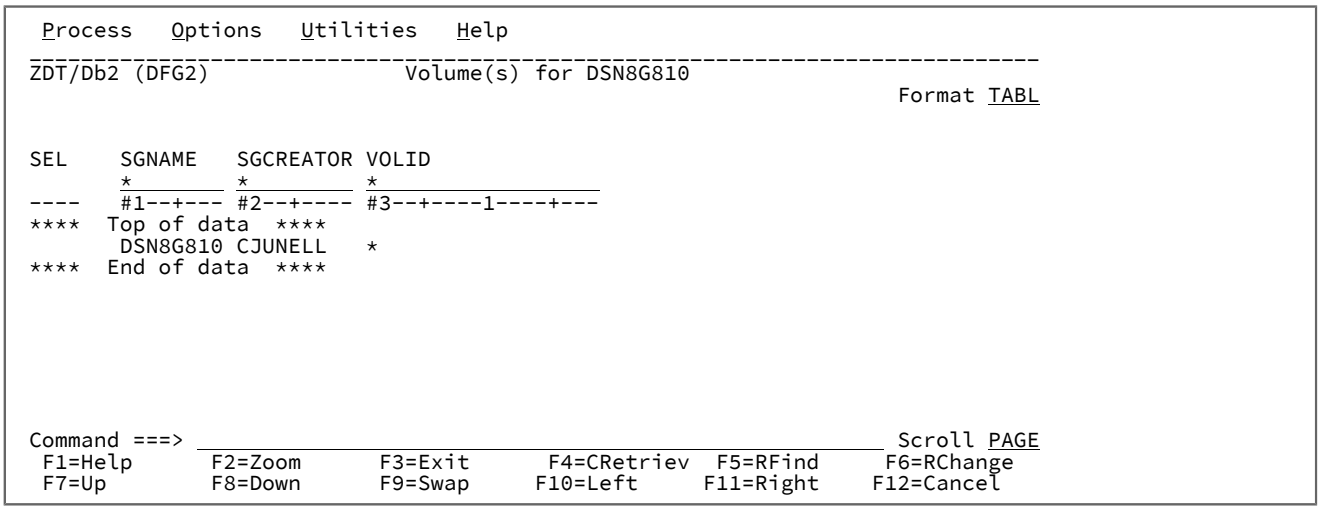
- Column Attributes panel (alphanumeric) on page 484
- Column Attributes panel (numeric) on page 490

Related tasks

- Specifying and editing a value list on page 112
- Scrambling data on page 292

Volumes panel

Figure 144. Volumes panel



Note:

To view other columns on this panel, press the Right function key (F11).

For a complete list of columns, refer to *"Db2® catalog tables"* in the *DB2 for z/OS SQL Reference* relevant to your version of Db2®.

SEL

Line command area.

For a list of the line commands you can use on this panel, see Table 22: Object list line commands on page 315.

Parent panels

- Storage Group(s) panel on page 848

Child panels

To display this panel...	Use/do this
	Line command ROW
Display Row panel on page 637	

Related tasks

- Working with object list panels on page 307
- Using the line command area (Cmd) on page 314

Related tasks

- Managing tables used by SQL explain on page 430

Chapter 18. ZDT/Db2 commands

This chapter describes the syntax and parameters for the primary commands that you can use in various ZDT/Db2 panels.

You enter primary commands on the command line of a panel.

Where the syntax allows you to specify more than one operand, you can use either a blank or a comma as a separator between each operand.

For example, both of the following commands are correct:

```
SORT col_num1 col_num2
SORT col_num1,col_num2
```

ABOUT primary command

The ABOUT command displays the current ZDT/Db2 version number and the PTF number of the Z Data Tools Db2® component in a window. The popup also shows the copyright information and any notes that are shipped with the product.

Syntax



Availability

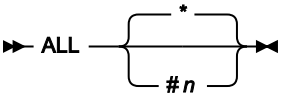
Available on all panels

Related tasks

Checking your ZDT/Db2 version on page 27

ALL primary command

Use the ALL primary command to select all columns from all, or one, of the tables you have specified on the primary Basic SELECT Prototyping panel.



Parameters

Asterisk (*)

Selects all columns from all the specified tables.

#n

Selects all columns from the table associated with correlation name *#n*.

Availability

Basic SELECT Prototyping panel on page 476

Related tasks

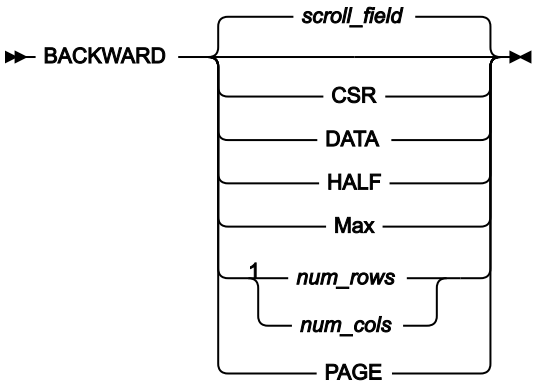
Using basic SQL prototyping on page 390

BACKWARD primary command

The BACKWARD primary command scrolls backwards (up) through your data.

The amount (number of rows or columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax



Notes:

[1] *num_rows* available in TABL display format; *num_cols* available in SNGL display format.

scroll_field

Scroll backwards by the amount indicated in the **Scroll field**. This is the default value if no parameter is used.

CSR

Scroll backwards to the cursor position.

DATA

Scroll backwards one row (TABL) or column (SNGL) less than a page of data.

HALF

Scroll backwards half a page of data.

Max

Scroll to top of the table. This has the same effect as TOP command.

num_rows

Scroll backwards *num_rows* rows. Valid range: 1–9999. Available in TABL display format.

num_cols

Scroll backwards *num_cols* columns. Valid range: 1–9999. Available in SNGL display format.

PAGE

Scroll backwards one page of data.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868

Related tasks

- Viewing and changing Db2 data on page 122

BOTTOM primary command

The BOTTOM primary command scrolls to the last page of data.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Viewing and changing Db2 data on page 122
- Scrolling on page 160

BROWSE primary command

The BROWSE primary command starts a new ZDT/Db2 Browse editor session.

Syntax

Figure 145. For editor panels

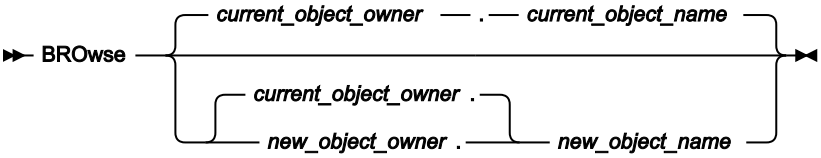


Figure 146. For Primary Option Menu panel



current_object_owner

The name of the owner of the object in the current ZDT/Db2 editor session.

current_object_name

The name of the object in the current ZDT/Db2 editor session.

new_object_owner

The name of the owner of the object to be pre-filled in the entry panel for the new ZDT/Db2 editor session.

new_object_name

The name of the object to be pre-filled in the entry panel for the new ZDT/Db2 editor session.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Primary Option Menu panel on page 760

Related tasks

- Viewing and changing Db2 data on page 122

CANCEL primary command

The CANCEL primary command ends the current ZDT/Db2 editor session without saving any changes.

In an editor session, the CANCEL primary command ends the current ZDT/Db2 editor session and any dependent related sessions. In Edit only, if you have made changes to your data, you are asked to confirm that you want to cancel your Edit session. Cancellation of an Edit session causes any uncommitted changes to the data (in the current or in a dependent related session) to be lost.

Note that ZDT/Db2 may issue Db2® commit points during an Edit session, depending on the current setting of the **Enter Key Processing** and **Commit Options** options (see Editor Options (6 of 8) panel on page 672). If commit points have been issued, CANCEL will only discard changes made since the last commit point.

If you have selected the Edit Commit Option **Whenever Enter is pressed**, the Db2® table is updated whenever you press Enter after having made a change. CANCEL in this situation does not reverse any changes made before the last time you pressed Enter.

QUIT is a synonym for CANCEL.

Syntax



Availability

Available on most ZDT/Db2 panels

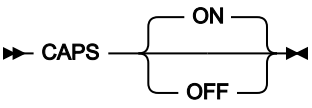
Related tasks

- Viewing and changing Db2 data on page 122
- Working with templates on page 74
- Working with SQL statements on page 390

CAPS primary command

The CAPS command determines whether data in the data area is to be translated to uppercase on input (CAPS ON), or whether it is to be left alone (CAPS OFF). The CAPS command overrides any previously issued CASE command. By default, lowercase data is not translated to uppercase.

Syntax



CAPS ON

Translates all data in rows that are changed by overtyping from lowercase to uppercase, and translates replacement data specified in the CHANGE command from lowercase to uppercase.

CAPS ON only affects rows changed after the CAPS ON command is entered.

CAPS OFF

Does not translate changed records to uppercase.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

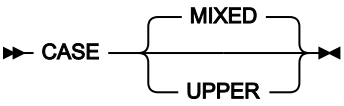
Related tasks

- Installation settings that affect CAPS on page 180
- Viewing and changing Db2 data on page 122

CASE primary command

The CASE command determines whether data in the data area is translated to uppercase on input (CASE UPPER), or whether it is left unchanged (CASE MIXED). The CASE command overrides any previously issued CAPS command. By default, lowercase data is not translated to uppercase.

Syntax



CASE MIXED

Does not translate changed or new rows to uppercase.

CASE UPPER

Translates all new rows from lowercase to uppercase, and translates all data in changed rows from lowercase to uppercase, regardless of how the rows are changed.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Installation settings that affect CAPS on page 180
- Viewing and changing Db2 data on page 122

CHANGE primary command

In a Z Data Tools/Db2 editor session, use the CHANGE command to locate and replace one or more occurrences of a character string or numeric value in the data being edited.

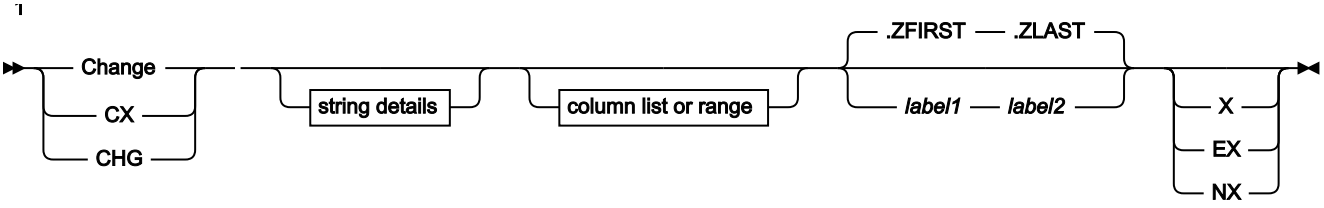
Note: If you limit the number of rows loaded into the ZDT/Db2 editor (by either specifying a row limit or a WHERE clause), the CHANGE command applies only to the data in the ZDT/Db2 editor session.

To find a string without changing it, use the FIND primary command.

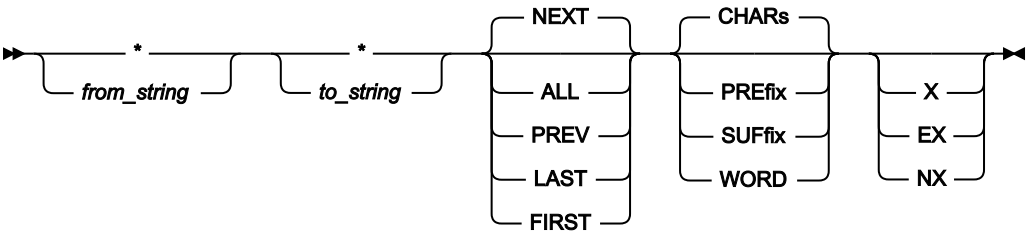
If you have zoomed in on a row, the CHANGE command affects only that row.

If the CHANGE command changes data, ZDT/Db2 places the cursor at the end (when scanning forward), or start (when scanning backwards), of the changed string or numeric data type column. If necessary, ZDT/Db2 scrolls to bring the string into view. If the changed data is in an excluded row, the excluded row is displayed.

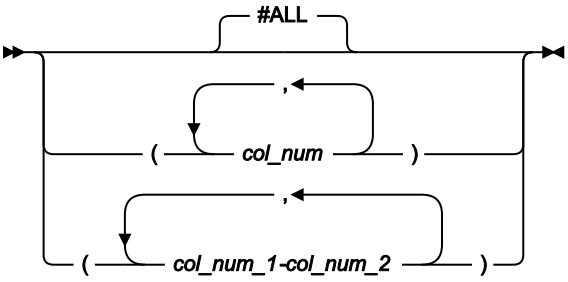
Syntax



string details



column list or range



Notes:

[1] You can specify parameters in any order.

*** (asterisk)**

When used in place of the search string, uses the search string specified on the previous CHANGE command as the search string.

When used in place of the replacement string, uses the replacement string specified on the previous CHANGE command as the replacement string.

The position of the * is important for CHANGE. If it is positioned before a string, it indicates the previous search argument; that is, it is treated as the first string. If a string is found prior to the *, then it is treated as the second string (the change argument), taking the change argument from the previous CHANGE command. To use both the previous search string and the previous change string, specify `CHANGE * *`.

from_string

The string you want to search for. The string, which can be up to 100 characters in length, can be:

- A character string not starting or ending with a single quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, the following command changes the strings `black`, `Black`, and `BLACK`:

  ```
  CHANGE black white
  ```

- A character string enclosed in single quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe a type of string to be found rather than the exact characters. It can contain blanks, alphabetic and numeric characters which represent themselves, or any of the special characters listed here, each of which represents a class of characters:

  =

  Any character.

  @

  Alphabetic characters.

  #

  Numeric characters.

  $

  Special characters.

  & notsym;

  Non-blank characters.

  .

  Invalid characters.

  -

  Non-numeric characters.

  <

  Lowercase alphabetics.

  >

  Uppercase alphabetics.

- X followed by a hexadecimal string enclosed in single quotation marks (`X'C1C2'`).
- R preceded or followed by a regular expression string enclosed in single or double quotation marks to specify a pattern for the string, rather than the exact characters. The string must conform to the format

allowed by the C runtime library regcomp function supported by the C runtime library and the C runtime library must be available. Use RC to perform a case sensitive search.

Examples of regular expressions are as follows:

'cat' or 'hat' in columns 1 to 10

```
r'[ch]at' 1 10
```

'payer' or 'payee'

```
r'pay(er!ee)'
```

'band' or 'bald' but not 'bad'

```
'ba.d'r
```

'feed' or 'weed' but not 'fed' or 'wed'

```
r'[fw]e{2}d'
```

'The' or 'She' but not 'the' or 'she'

```
rc'[TS]he'
```

The SETLOCALE function is used to define the locale used by the regcomp function to compile the regular expression. The locale used for the SETLOCALE function is based on the terminal CCSID. Locales are supported for the following terminal CCSIDs:

```
00037   00871   01123   01156
00273   00875   01140   01157
00277   00924   01141   01158
00278   00930   01142   01160
00280   00933   01143   01165
00284   00935   01144   01364
00285   00937   01145   01371
00290   00939   01146   01388
00297   01025   01147   01390
00424   01026   01148   01399
00425   01027   01149   04971
00500   01047   01153   05123
00838   01112   01154   08482
00870   01122   01155   12712
```

If you are using a terminal with a different CCSID, then the default C locale is used when compiling the regular expression.

- A numeric value (only when you limit the search by specifying column numbers, and only when the column being searched is a numeric column).

to_string

The string you want to replace *from_string*. The string, which can be up to 100 characters in length, can be:

- A character string not starting or ending with a single quotation mark and not containing any embedded blanks or commas. If CAPS ON or CASE UPPER is in effect, then *to_string* is converted to uppercase.
- A character string enclosed in single quotation marks. The string can contain blanks and commas. If CAPS ON or CASE UPPER is in effect, then *to_string* is converted to uppercase. The string can be a null string (`''`).

- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `c` ('Frog'C). The string can contain blanks and commas. Case is respected and retained. The string can be a null string (`C''`).
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe the change to be made. You can change characters from uppercase to lowercase or from lowercase to uppercase, or leave the field the same using these string special characters.

  =

  Any character.

  <

  Lowercase alphabetics.

  >

  Uppercase alphabetics.

  Examples of picture strings used with the CHANGE command:

  `c p'<' p'>' #3`

  Uppercase characters in column 3.

  `c p'>' '¬' #2`

  Change any uppercase to "not sign" column 2 (note: "not sign" has no special meaning in a "to" string).

  `c p'<#' p'>=' #1`

  Change any lowercase, any number to uppercase, the number found in column 1.

  When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process.

  .
- X followed by a hexadecimal string enclosed in single quotation marks (`X'C1C2'`). Case is respected and retained (the hexadecimal values are used exactly as specified).
- A numeric value.

NEXT

This is the default setting. Causes the search to begin at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches ahead to find the next occu rrence of the string. If the next occurrence of the string is in an excluded row and you do not limit the search to non-excluded rows, only the first excluded row containing the string is shown.

ALL

Causes the search to begin at the top of the data and find and replace all occurrences of the string. If you do not limit the search to non-excluded rows, the string is replaced in all rows (excluded and non-excluded). Any excluded rows affected by the change are redisplayed.

FIRST

Causes the search to begin at the beginning of the table and search ahead.

LAST

Causes the search to begin at the end of the table and search backwards.

PREV

Causes the search to begin at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches backwards to find the string.

CHARS

Matches the search string anywhere in the data.

PREFIX

Matches the search string wherever it appears as a prefix in the data. To be a prefix, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by an alphanumeric character.

SUFFIX

Matches the search string wherever it appears as a suffix in the data. To be a suffix, the matched text must be preceded by an alphanumeric character, and must be followed by a non-alphanumeric character or be the end of a line or field.

WORD

Matches the search string wherever it appears as a word in the data. To be a word, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by a non-alphanumeric character or be the end of a line or field.

#ALL

Each column is searched according to its template attributes.

col_num

The Db2® column number (specified as *#n*) of a column to be included in the data search. Multiple column numbers must be separated by a comma or, if enclosed in brackets, separated by a blank or comma.

col_num_1

The first field reference of a range of fields. It cannot be subscripted. If the *col_num_1* field reference value is less than the lowest displayed field reference value, the lowest displayed field reference value is used.

col_num_2

The last field reference of a range of fields. It cannot be subscripted. If the *col_num_2* field reference value is greater than the highest displayed field reference value, the highest displayed field reference value is used.

The *col_num_1* and *col_num_2* field reference values must be separated by a hyphen (-). Spaces are permitted between the hyphen and the field reference values.

If *col_num_1* is a higher value than *col_num_2*, the search process reverses the operands.

label1

Label identifying the start of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2

Label identifying the end of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

EX

Excluded rows only.

NX

Non-excluded rows only.

X

Same as EX.

Availability

- Table Edit panel on page 868

Related tasks

- Changing data with the CHANGE command on page 182

COL primary command

When you use the SQL prototyping, execution and analysis option (4), you can use the COL primary command:

- On the Plan Table Rows panel to switch panel format to show the collection ID.
- On the Statement Table Rows panel to switch panel format to show the collection ID.

Syntax



Availability

Available on all panels Plan Table Rows panel on page 754

Related tasks

Working with SQL statements on page 390

CHARPOS primary command

In SNGL display mode only, the CHARPOS command displays, or hides, the *"character position range"* for long character or graphic columns. The command is used to dynamically turn on or off the *"column position range"* information.

Syntax



The CHARPOS command is a *toggle* command. Each time you enter the command the display status of the *"column position range"* is reversed.

When displayed, the *"character position range"* information is displayed below the column name for the second and subsequent rows used to display data for long columns. Here is an example:

```
Column            Data
HIGH2KEY          (this row has data for chars 1-103 of column HIGH2KEY)
  104-206         (this row has data for chars 104-206 of column HIGH2KEY)
  207-309         (this row has data for chars 207-309 of column HIGH2KEY)
```

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

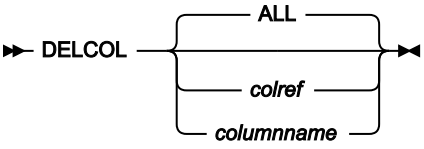
Related tasks

- Locating a column on page 142
- Locating a row or column in a list of Db2 objects on page 314

DELCOL primary command

In the row selection criteria panel of the Z Data Tools template editor, the DELCOL command removes any column expression entered against one or all columns. Any affected column name and, when displayed, data type fields that are editable are changed to protected. You can also enter the DC prefix command against a column to achieve the same result.

Syntax



colref

A reference to a single column, for example, #4.

columnname

The name of a column in the Db2® object or result table being processed.

DELCOL with no parameters deletes all column expressions and changes all column name fields and, when displayed, data type fields to protected.

DELCOL with either a column reference or a column name removes any column expression for the specified column and changes the specified column name field and, when displayed, data type fields to protected.

Availability

- Row Selection Criteria panel on page 799

Related tasks

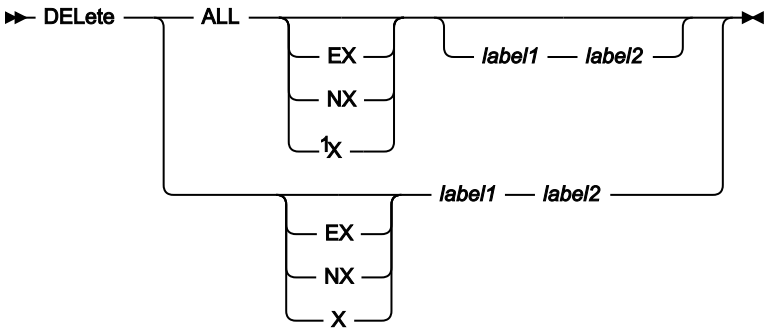
- Row selection by column on page 86

DELETE primary command

The DELETE primary command deletes rows from the data being edited.

Syntax

Figure 147. TABL display format



Notes:

[1] If none of these parameters (EX, NX, or X) are specified, then the DELETE command applies to both excluded and non-excluded rows.

Figure 148. SNGL display format



ALL

Causes the DELETE command to delete all rows (or all excluded or non-excluded rows) within the specified range of rows (or within the entire table, if you do not specify a range).

If you do not specify the ALL parameter, then the DELETE command deletes only the first row (or only the first excluded or non-excluded row) within the specified range.

EX

Excluded rows only.

NX

Non-excluded rows only.

X

Same as EX.

label1

Label identifying the start of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter “Z” indicate an editor-assigned label.

label2

Label identifying the end of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter “Z” indicate an editor-assigned label.

Availability

- Table Edit panel on page 868

Related tasks

- Deleting rows on page 191

DESCRIBE primary command

The DESCRIBE primary command displays the Db2® Template Description panel where you can enter the required description for the template.

Syntax



Availability

- Column Selection/Edit panel on page 497

Related tasks

- Editing a template on page 84

DESELECT primary command

In advanced SELECT prototyping, if you enter the DESELECT primary command and the cursor is not currently placed on an element in the currently displayed fragment, the currently displayed fragment and its dependent fragments are removed from the syntax, and control returns to the parent of the currently displayed fragment. If the currently displayed fragment is the top fragment, this has the same effect as CANCEL or END.

If you enter the DESELECT primary command with the cursor placed on a fragment field in the currently displayed fragment, the selected fragment and its dependent fragments are removed from the syntax, and control remains in the currently displayed fragment.

If the deselected fragment is the last selected (or only) item in a repeat instance, the entire instance is deleted.

Syntax



Availability

Advanced SELECT Prototyping panel on page 451

Related tasks

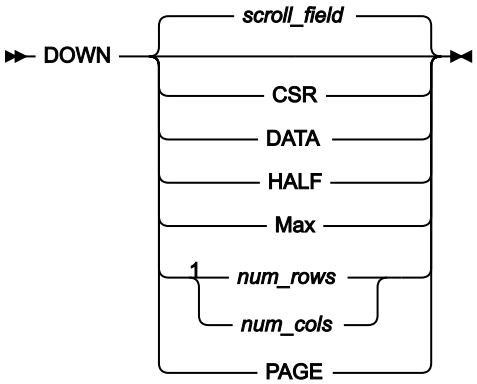
Using advanced SQL prototyping on page 399

DOWN primary command

The DOWN primary command scrolls down (forward) through your data.

The amount (number of rows or columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax



Notes:

[1] *num_rows* available in TABL display format; *num_cols* available in SNGL display format.

scroll_field

Scroll down by the amount indicated in the **Scroll field**. This is the default value if no parameter is used.

CSR

Scroll down to the cursor position.

DATA

Scroll down one row (TABL) or column (SNGL) less than a page of data.

HALF

Scroll down half a page of data.

Max

Scroll to bottom of table. This has the same effect as the BOTTOM command.

num_rows

Scroll down *num_rows* rows. Valid range: 1–9999. Available in TABL display format.

num_cols

Scroll down *num_cols* columns. Valid range: 1–9999. Available in SNGL display format.

PAGE

Scroll down one page of data.

Examples

`DOWN 8`

Scrolls down 8 lines.

`DOWN`

Scrolls down the number of lines indicated by the **Scroll** field.

`DOWN M`

Scrolls down to the last page of data.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Enter, Execute and Explain SQL Statements panel on page 684

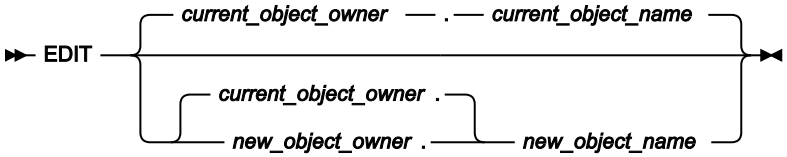
Related tasks

- Viewing and changing Db2 data on page 122
- Scrolling on page 160
- Entering, executing, and explaining SQL statements on page 424

EDIT primary command

The EDIT primary command starts a new ZDT/Db2 Edit session.

Syntax



current_object_owner

The name of the owner of the object in the current ZDT/Db2 editor session.

current_object_name

The name of the object in the current ZDT/Db2 editor session.

new_object_owner

The name of the owner of the object to be pre-filled in the entry panel for the new ZDT/Db2 editor session.

new_object_name

The name of the object to be pre-filled in the entry panel for the new ZDT/Db2 editor session.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

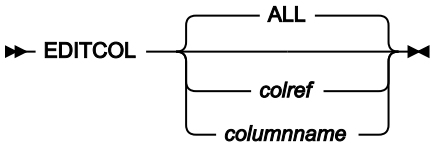
Related tasks

- Viewing and changing Db2 data on page 122

EDITCOL primary command

In the row selection criteria panel of the Z Data Tools template editor, the EDITCOL command changes the column name and, when displayed, the data type fields from protected to editable. This command allows a column name expression to be entered. You can also enter the EC prefix command against a column to achieve the same result.

Syntax



colref

A reference to a single column, for example, #4.

columnname

The name of a column in the Db2® object or result table being processed.

EDITCOL with no parameters changes all column name fields and, when displayed, data type fields from protected to editable.

EDITCOL with either a column reference or column name changes the specified column name field and, when displayed, data type fields from protected to editable.

Availability

- Row Selection Criteria panel on page 799

Related tasks

- Row selection by column on page 86

END primary command

END is a pseudonym for FILE.

- In View or Browse, the END primary command:
  - Ends the View or Browse editor session.
- In Edit, the END primary command:
  - Saves any uncommitted changes and ends the Edit session.
  - Fails if Db2® reports errors when ZDT/Db2 attempts to save changes made in the Edit session, or there are dependent edit sessions of related tables.
- When you edit a template, the END primary command:
  - Displays the save panel where you can save the template as another name.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868

- Table View panel on page 882
- Column Selection/Edit panel on page 497

Related tasks

- Viewing and changing Db2 data on page 122
- Working with templates on page 74

EXCLUDE primary command

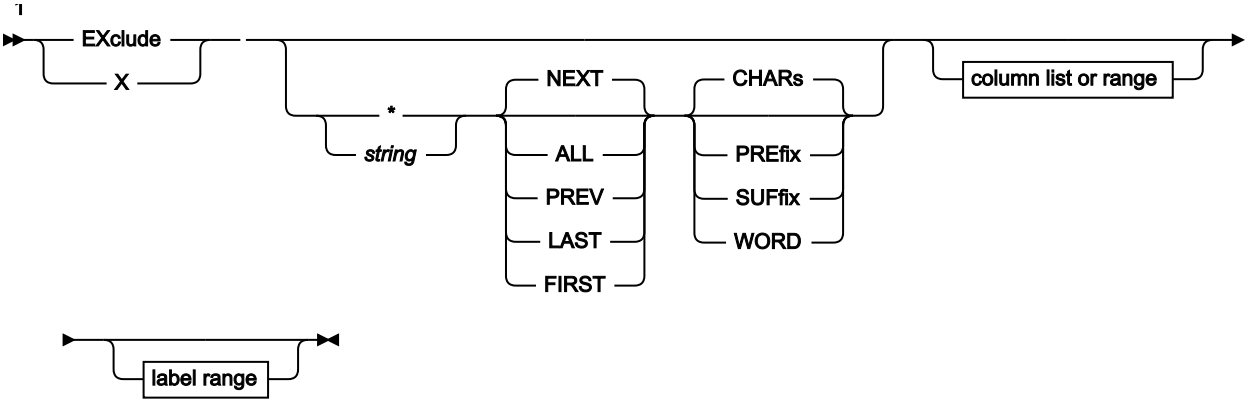
The EXCLUDE primary command excludes from display certain rows in the data being viewed, browsed, or edited.

You can only use the EXCLUDE primary command in TABL display format.

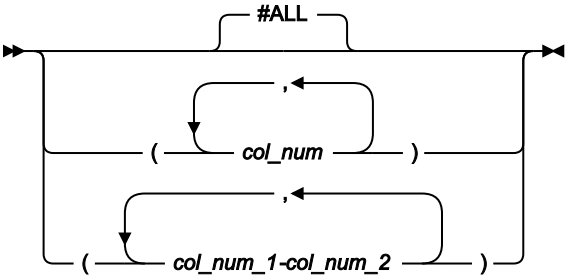
To redisplay excluded rows, enter the command:

```
RESET
```

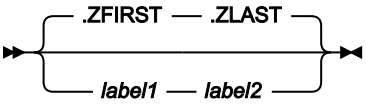
Syntax



column list or range



label range



Notes:

[1] You can specify parameters in any order.

*** (asterisk)**

Uses the search string specified on the previous EXCLUDE command as the search string.

string

The search string you want to search for. Rows containing this string, within the limits imposed by the other EXCLUDE command parameters, are excluded. The string can be:

- A character string not starting or ending with a single quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, "Mixed string" matches "MIXED string".
- A character string enclosed in single quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, "Exact string" matches "exact string".

- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `c` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, “Exact string” does not match “exact string”.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe a type of string to be found rather than the exact characters. It can contain blanks, alphabetic and numeric characters which represent themselves, or any of the special characters listed here, each of which represents a class of characters:

  =

  Any character.

  @

  Alphabetic characters.

  #

  Numeric characters.

  $

  Special characters.

  & notsym;

  Non-blank characters.

  .

  Invalid characters.

  -

  Non-numeric characters.

  <

  Lowercase alphabetics.

  >

  Uppercase alphabetics.

Examples of picture strings used with the EXCLUDE command:

```
ex p'.' #4
```

Exclude invalid character in column 4.

```
ex p'###'
```

Exclude 3-digit number.

```
ex '@1'p #1
```

Exclude label a1,b1,c1, (and so on) in column 1.

ex p'<'

Exclude the next lowercase alphabetic character.

ex p'¬' #2

Exclude the next non-blank in column 2.

When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process.

- X followed by a hexadecimal string enclosed in single quotation marks For example, X'C1C2'.
- R preceded or followed by a regular expression string enclosed in single or double quotation marks to specify a pattern for the string, rather than the exact characters. The string must conform to the format allowed by the C runtime library regcomp function supported by the C runtime library and the C runtime library must be available. Use RC to perform a case sensitive search.

Examples of regular expressions are as follows:

'cat' or 'hat' in columns 1 to 10

```
r'[ch]at' 1 10
```

'payer' or 'payee'

```
r'pay(er!ee)'
```

'band' or 'bald' but not 'bad'

```
'ba.d'r
```

'feed' or 'weed' but not 'fed' or 'wed'

```
r'[fw]e{2}d'
```

'The' or 'She' but not 'the' or 'she'

```
rc'[TS]he'
```

The SETLOCALE function is used to define the locale used by the regcomp function to compile the regular expression. The locale used for the SETLOCALE function is based on the terminal CCSID. Locales are supported for the following terminal CCSIDs:

```
00037   00871   01123   01156
00273   00875   01140   01157
00277   00924   01141   01158
00278   00930   01142   01160
00280   00933   01143   01165
00284   00935   01144   01364
00285   00937   01145   01371
00290   00939   01146   01388
00297   01025   01147   01390
00424   01026   01148   01399
00425   01027   01149   04971
00500   01047   01153   05123
00838   01112   01154   08482
00870   01122   01155   12712
```

If you are using a terminal with a different CCSID, then the default C locale is used when compiling the regular expression.

- A numeric value (only when you limit the search by specifying column numbers, and only when the column being searched is a numeric column).

NEXT

This is the default setting. Causes the search to begin at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches ahead to find the next occu rrence of *string* in a row that is not already excluded.

ALL

Causes the search to begin at the top of the data and find all occurrences of *string*. If not specified, the search begins at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches ahead to find the next occurrence of *string* in a row that is not already excluded.

The command EXCLUDE ALL excludes all displayed rows.

FIRST

Causes the search to begin at the beginning of the table and search ahead.

LAST

Causes the search to begin at the end of the table and search backwards.

PREV

Causes the search to begin at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches backwards to find the string.

CHARS

Matches the search string anywhere in the data.

PREFIX

Matches the search string wherever it appears as a prefix in the data. To be a prefix, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by an alphanumeric character.

SUFFIX

Matches the search string wherever it appears as a suffix in the data. To be a suffix, the matched text must be preceded by an alphanumeric character, and must be followed by a non-alphanumeric character or be the end of a line or field.

WORD

Matches the search string wherever it appears as a word in the data. To be a word, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by a non-alphanumeric character or be the end of a line or field.

#ALL

Each column is searched according to its template attributes.

col_num

The Db2® column number (specified as *#n*) of a column to be included in the data search. Multiple column numbers must be separated by a comma or, if enclosed in brackets, separated by a blank or comma.

col_num_1

The first field reference of a range of fields. It cannot be subscripted. If the *col_num_1* field reference value is less than the lowest displayed field reference value, the lowest displayed field reference value is used.

col_num_2

The last field reference of a range of fields. It cannot be subscripted. If the *col_num_2* field reference value is greater than the highest displayed field reference value, the highest displayed field reference value is used.

The *col_num_1* and *col_num_2* field reference values must be separated by a hyphen (-). Spaces are permitted between the hyphen and the field reference values.

If *col_num_1* is a higher value than *col_num_2*, the search process reverses the operands.

label1

Label identifying the start of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2

Label identifying the end of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

Availability

- Table Browse panel on page 866
- Table View panel on page 882
- Table Edit panel on page 868

Related tasks

- Excluding rows on page 162

EXECUTE primary command

You use the EXECUTE primary when you are working with SQL statements.

For basic SELECT prototyping, the EXECUTE command executes the SELECT statement.

Note: Before you use the EXECUTE primary command, you must have selected at least one column.

For advanced SELECT prototyping, the EXECUTE primary command executes the prototyped command if it is syntactically complete. ZDT/Db2 displays a result table similar to the Select Statement Browse panel.

When you use the Enter, Execute and Explain SQL Statements panel, the EXECUTE primary command executes the SQL statements in the statement entry area. The EXECUTE primary command is executed by default when you press Enter and the command line is blank.

Syntax



Availability

- Basic SELECT Prototyping panel on page 476
- Advanced SELECT Prototyping panel on page 451
- Enter, Execute and Explain SQL Statements panel on page 684

Related tasks

- Using basic SQL prototyping on page 390
- Using advanced SQL prototyping on page 399
- Entering, executing, and explaining SQL statements on page 424

EXPAND primary command

The EXPAND primary command displays a panel for inputting or editing long row selection criteria. You use this command from the Column Selection/Edit panel (you must position the cursor on the **Row Selection Criteria** input field).

Syntax



Availability

- Column Selection/Edit panel on page 497

Related tasks

- Selecting rows on page 86

EXPLAIN primary command

The EXPLAIN primary command explains the SQL statements in the statement entry area. ZDT/Db2 builds an SQL EXPLAIN statement using the contents of the statement entry area to build the FOR clause. If applicable, ZDT/Db2 also builds a SET QUERYNO clause using the contents of the **Query number** option or a generated value.

When you use the EXPLAIN primary command, ZDT/Db2 shows the information obtained by the generated SQL EXPLAIN statement in the Plan Table Rows panel.

Syntax



Availability

- Enter, Execute and Explain SQL Statements panel on page 684

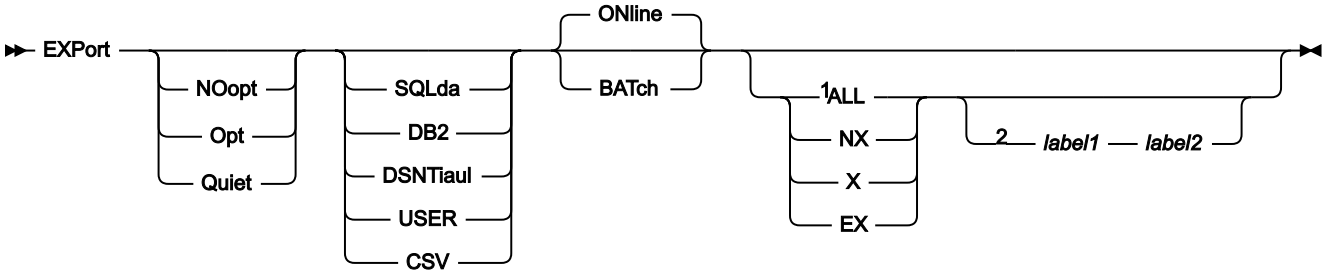
Related tasks

- Entering, executing, and explaining SQL statements on page 424

EXPORT primary command

The EXPORT primary command starts a dialog that allows either the result table currently shown in the ZDT/Db2 editor session, or rows in the current ZDT/Db2 editor session, to be exported to a data set.

Syntax



Notes:

[1] NX, X, EX valid for TABL format only.
[2] Label range valid for TABL format only.

NOOPT

Overrides the current setting for the **Show export options** option on the Editor Options (4 of 8) panel. The Export Options panel is not displayed.

The data format for the export is determined as follows: The current value for the export data format is used, unless a data format parameter is specified in the EXPORT command. In the latter case, the parameter specified in the EXPORT command overrides the current export data format value.

OPT

Overrides the current setting for the **Show export options** option on the Editor Options (4 of 8) panel. The Export Options panel is displayed.

QUIET

Overrides the current setting for the **Show export options** option on the Editor Options (4 of 8) panel. The Export Options panel is not displayed. This option also suppresses the display of the Export "To" panel. The current values specified on the Export "To" panel are used to process the EXPORT command.

SQLDA

Data is exported in ZDT/Db2 internal (SQLDA) format. This is equivalent to specifying 1 for the **Data Format** option on the Export Options (1 of 3) panel.

Db2®

Data is exported in Db2® unload format. This is equivalent to specifying 2 for the **Data Format** option on the Export Options (1 of 3) panel.

DSNTIAUL

Data is exported in same format as that used by the DSNTIAUL sample program. This is equivalent to specifying 3 for the **Data Format** option on the Export Options (1 of 3) panel.

USER

Data is exported in a user-defined data format. This is equivalent to specifying 4 for the **Data Format** option on the Export Options (1 of 3) panel.

CSV

Data is exported in a comma-separated variables (delimited) format. This is equivalent to specifying 5 for the **Data Format** option on the Export Options (1 of 3) panel.

ONLINE

The export is performed on-line.

BATCH

Generate a JCL deck to execute the export for the Db2® object or SQL SELECT statement in batch mode. You cannot specify BATCH with any of ALL|NX|X|EX.

ALL

Copies all rows in the current ZDT/Db2 editor session to the target data set.

NX

Copies all non-excluded rows in the current ZDT/Db2 editor session to the target data set.

X

Copies all excluded rows in the current ZDT/Db2 editor session to the target data set.

EX

Same as X.

label1

Label identifying the start of a range of rows to be processed. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2

Label identifying the end of a range of rows to be processed. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Excluding rows on page 162

FILE primary command

FILE is a pseudonym for END.

- In View or Browse, the FILE primary command:
  - Ends the View or Browse session.
- In Edit, the FILE primary command:
  - Saves any uncommitted changes and ends the Edit session.
  - Fails if Db2® reports errors when ZDT/Db2 attempts to save changes made in the Edit session, or there are dependent edit sessions of related tables.
- When you edit a template, the FILE primary command:
  - Displays the save panel where you can save the template as another name.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Column Selection/Edit panel on page 497

Related tasks

- Viewing and changing Db2 data on page 122
- Working with templates on page 74

FIND primary command

The FIND primary command searches the data for a character string or a numeric field with a particular numeric value.

In a Z Data Tools/Db2 editor session, or the Object List utility:

- If the string or numeric value is found, then the FIND command positions the cursor at the beginning of the string or numeric field and, if necessary, automatically scrolls the found data into view.
- The FIND command highlights all occurrences of the search string or numeric value (even when you specify FIRST, LAST, PREV, NEXT, EX, NX, or X). To turn off the highlighting, enter the RESET FIND command.

In Print Browse:

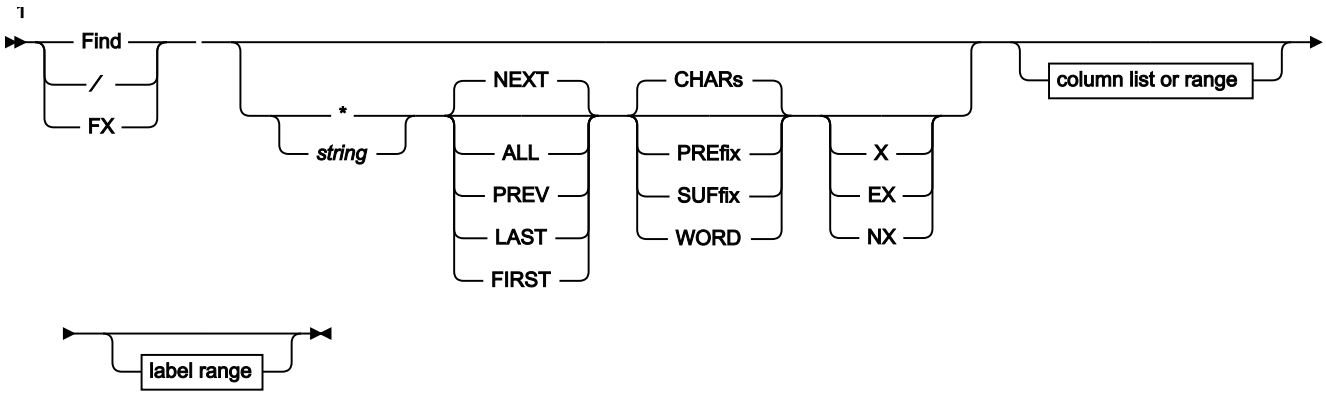
- If the character string is found, the record containing the string is displayed at the top position.

To find the next occurrence of the same string, press the RFind function key (F5), enter FIND *, or enter the FIND command with no parameters. A message is displayed if the string cannot be found.

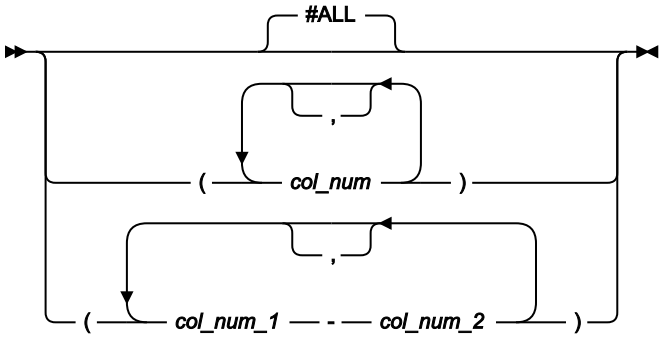
Note: `FIND *` does *not* repeat the previous FIND command with the same parameters. `FIND *` repeats the previous FIND command with the same string argument, but all other parameters revert to their default values unless specified.

Syntax

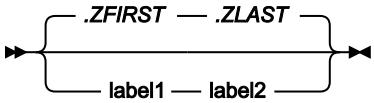
Figure 149. Editor session



column list or range (SNGL or TABL display format)



label range



Notes:

[1] You can specify parameters in any order.

Figure 150. Object List utility

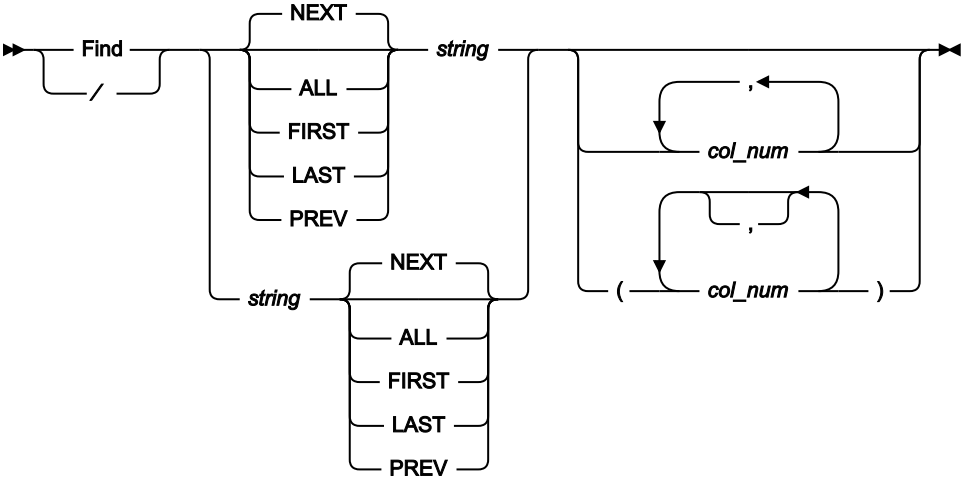
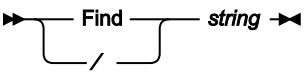


Figure 151. Print Browse



*** (asterisk)**

Uses the search string specified on the previous FIND command as the search string.

string

The string you want to search for. The string, that can be up to 100 characters in length, can be:

- A character string not starting or ending with a single quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, the following command finds the strings `black`, `Black`, and `BLACK`:

  ```
  FIND black
  ```

- A character string enclosed in single quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe a type of string to be found rather than the exact characters. It can contain blanks, alphabetic and numeric characters which represent themselves, or any of the special characters listed here, each of which represents a class of characters:

  =

  Any character.

@

Alphabetic characters.

#

Numeric characters.

$

Special characters.

& notsym;

Non-blank characters.

.

Invalid characters.

-

Non-numeric characters.

<

Lowercase alphabetics.

>

Uppercase alphabetics.

Examples of picture strings used with the FIND command:

`find p'.' #3`

Find invalid character in columns 3.

`find p'###'`

Find 3 digit number (for example, 101 but not 99).

`find '@1'p #1`

Find label a1,b1,c1, (and so on) in column 1.

`find p'<'`

Find lowercase alphabetic character.

`find p'¬' #2`

Find non-blank character in column 2.

When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process.

- X followed by a hexadecimal string enclosed in single quotation marks. For example, `X'C1C2'`.
- R preceded or followed by a regular expression string enclosed in single or double quotation marks to specify a pattern for the string, rather than the exact characters. The string must conform to the format

allowed by the C runtime library regcomp function supported by the C runtime library and the C runtime library must be available. Use RC to perform a case sensitive search.

Examples of regular expressions are as follows:

'cat' or 'hat' in columns 1 to 10

```
r'[ch]at' 1 10
```

'payer' or 'payee'

```
r'pay(er!ee)'
```

'band' or 'bald' but not 'bad'

```
'ba.d'r
```

'feed' or 'weed' but not 'fed' or 'wed'

```
r'[fw]e{2}d'
```

'The' or 'She' but not 'the' or 'she'

```
rc'[TS]he'
```

The SETLOCALE function is used to define the locale used by the regcomp function to compile the regular expression. The locale used for the SETLOCALE function is based on the terminal CCSID. Locales are supported for the following terminal CCSIDs:

```
00037   00871   01123   01156
00273   00875   01140   01157
00277   00924   01141   01158
00278   00930   01142   01160
00280   00933   01143   01165
00284   00935   01144   01364
00285   00937   01145   01371
00290   00939   01146   01388
00297   01025   01147   01390
00424   01026   01148   01399
00425   01027   01149   04971
00500   01047   01153   05123
00838   01112   01154   08482
00870   01122   01155   12712
```

If you are using a terminal with a different CCSID, then the default C locale is used when compiling the regular expression.

- (ZDT/Db2 editor session only.) A numeric value (only when you limit the search by specifying column numbers, and only when the column being searched is a numeric column).

NEXT

This is the default setting. Causes the search to begin at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches ahead to find the next occu rrence of *string*.

ALL

Causes the search to begin at the top of the data and find all occurrences of the string.

Note: Not-selected rows that are hidden from display or represented by shadow lines are not processed by the FIND command, even when the ALL parameter is specified.

FIRST

Causes the search to begin at the beginning of the table and search ahead.

LAST

Causes the search to begin at the end of the table and search backwards.

PREV

Causes the search to begin at the cursor location (if the cursor is within the data portion of the display) or the beginning of the first row displayed, and searches backwards to find the string.

CHARS

Matches the search string anywhere in the data.

PREFIX

Matches the search string wherever it appears as a prefix in the data. To be a prefix, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by an alphanumeric character.

SUFFIX

Matches the search string wherever it appears as a suffix in the data. To be a suffix, the matched text must be preceded by an alphanumeric character, and must be followed by a non-alphanumeric character or be the end of a line or field.

WORD

Matches the search string wherever it appears as a word in the data. To be a word, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by a non-alphanumeric character or be the end of a line or field.

#ALL

Each column is searched according to its template attributes.

col_num

The Db2® column number (specified as *#n*) of a column to be included in the data search. Multiple column numbers must be separated by a comma or, if enclosed in brackets, separated by a blank or comma.

col_num_1

The first field reference of a range of fields. It cannot be subscripted. If the *col_num_1* field reference value is less than the lowest displayed field reference value, the lowest displayed field reference value is used.

col_num_2

The last field reference of a range of fields. It cannot be subscripted. If the *col_num_2* field reference value is greater than the highest displayed field reference value, the highest displayed field reference value is used.

The *col_num_1* and *col_num_2* field reference values must be separated by a hyphen (-). Spaces are permitted between the hyphen and the field reference values.

If *col_num_1* is a higher value than *col_num_2*, the search process reverses the operands.

label1

(ZDT/Db2 editor session only.) Label identifying the start of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2

(ZDT/Db2 editor session only.) Label identifying the end of a range of rows. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

EX

Excluded rows only.

NX

Non-excluded rows only.

X

Same as EX.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Any Print Browse display panel
- Collections panel on page 482
- Columns panel on page 503
- Databases panel on page 609
- Database Request Modules panel on page 607
- Distinct Types panel on page 637
- Functions panel on page 709
- Indexes panel on page 723
- Application Packages panel on page 469
- Application Plans panel on page 473
- Schemas panel on page 812
- Stored Procedures panel on page 852
- Synonyms panel on page 855
- Tables, Views and Aliases panel on page 891
- Table Spaces panel on page 879

- Triggers panel on page 903
- Storage Group(s) panel on page 848

Related tasks

- Viewing data in a Z Data Tools/Db2 editor session on page 141
- Looking at the print output from your ZDT/Db2 session on page 333
- Finding a string in a list of Db2 objects on page 314
- Locating a row or column in a list of Db2 objects on page 314

FMAP primary command

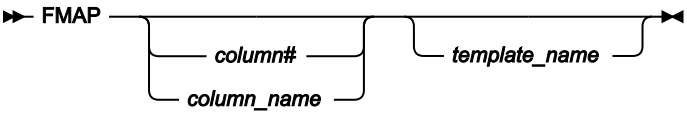
The FMAP command can be used to edit or view a column in a table using a template. If the column being mapped is shorter than the template mapping a warning message is issued when data is changed and any data which is beyond the end of the mapped field is ignored.

The row containing the column to be mapped must be at the top of the screen or the cursor may be positioned on the row when the FMAP command is issued.

If the column identification parameter is not specified and the cursor is positioned on a column that column is used. Otherwise a Field List Selection panel will be displayed.

If the template name is not specified a panel is displayed where this information can be supplied.

Syntax



column#

The reference number of the column to be mapped.

column_name

The name of the column to be mapped.

template_name

The name of the template to be used to map the column. A partially or fully qualified data set name together with the member name can be specified.

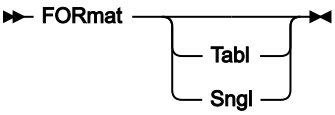
Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

FORMAT primary command

The FORMAT primary command toggles between tabular display format (TABL) and single-row display format (SNGL). It has the same effect as changing the **Format** field (in the upper right corner of the panel).

Syntax



Tabl

In SNGL display format, changes to TABL display format and positions the current row at the top of the panel. This parameter is optional.

Sngl

In TABL display format, changes to SNGL display format and displays the row previously shown at the top of the panel. This parameter is optional.

Note:

1. You can also use the FS and FT primary commands to toggle between TABL and SNGL display formats.
2. In View and Edit, when in TABL display format, you can use the FS *prefix* command against a row to change to SNGL display format and display that row.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

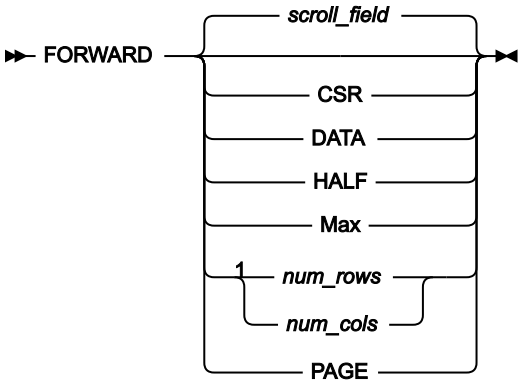
- Viewing and changing Db2 data on page 122

FORWARD primary command

The FORWARD primary command scrolls forward (down) through your data.

The amount (number of rows or columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax



Notes:

[1] *num_rows* available in TABL display format; *num_cols* available in SNGL display format.

scroll_field

Scroll forward by the amount indicated in the **Scroll field**. This is the default value if no parameter is used.

CSR

Scroll forward to the cursor position.

DATA

Scroll forward one row (TABL) or column (SNGL) less than a page of data.

HALF

Scroll forward half a page of data.

Max

Scroll to bottom of table. This has the same effect as the BOTTOM command.

num_rows

Scroll forward *num_rows* rows. Valid range: 1–9999. Available in TABL display format.

num_cols

Scroll forward *num_cols* columns. Valid range: 1–9999. Available in SNGL display format.

PAGE

Scroll forward one page of data.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

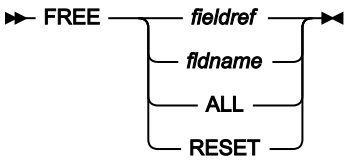
Related tasks

- Viewing and changing Db2 data on page 122

FREE primary command

In the ZDT/Db2 editor, the FREE command removes any scrolling lock on one or more columns. You can set scrolling locks with the HOLD editor primary command or by using the H prefix command in the template editor.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

Examples

`FREE ALL`

Removes all scrolling locks.

`FREE RESET`

Removes all scrolling locks.

`FREE #4`

Removes any scrolling lock from column #4.

`FREE #9 #2`

Removes any scrolling lock from columns #2 and #9.

`FREE #5-#9`

Removes any scrolling lock from columns #5, #6, #7, #8, and #9.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

FS primary command

In TABL display format, you can use the FS primary command to change to SNGL display format. The row currently displayed at the top of the panel in TABL display format is shown in SNGL display format.

The FS primary command has the same effect as changing the **Format** field (in the upper right corner of the panel) from *"TABL"* to *"SNGL"*.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Viewing and changing Db2 data on page 122

FT primary command

In SNGL display format, you can use the FT primary command to change to TABL display format. The row currently displayed in SNGL display format is shown at the top of the panel in TABL display format.

The FT primary command has the same effect as changing the **Format** field (in the upper right corner of the panel) from *"SNGL"* to *"TABL"*.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

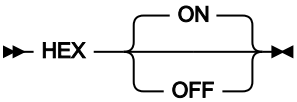
Related tasks

- Viewing and changing Db2 data on page 122

HEX primary command

The HEX primary command sets or resets the hexadecimal display format.

Syntax



(no parameter)

Toggles the correct hexadecimal-display status. That is, it turns hexadecimal display ON if currently OFF, and turns hexadecimal display OFF if currently ON.

OFF

Display the data in character format.

ON

Display the hexadecimal representation of the data vertically (three lines for each byte).

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

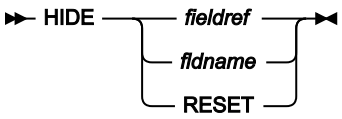
Related tasks

- Displaying data in hexadecimal format on page 165

HIDE primary command

In the ZDT/Db2 editor, the HIDE command removes a column from the data display. A hidden column can be displayed by using the SHOW editor primary command or by using the S prefix command in the template editor.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

Examples

`HIDE RESET`

Shows all hidden columns.

`HIDE #4`

Does not display column #4.

`HIDE #7 #1`

Does not display columns #1 and #7

`HIDE #2-#5`

Does not display columns #2, #3, #4, and #5.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

HINT primary command

When you use the SQL prototyping, execution and analysis option (4), you can use the HINT primary command:

- On the Plan Table Rows panel to switch panel format to show hint IDs and hints used
- On the Statement Table Rows panel to switch panel format to show hint IDs and hints used

Syntax



Availability

- Plan Table Rows panel on page 754
- Statement Table Rows panel on page 845

Related tasks

Working with SQL statements on page 390

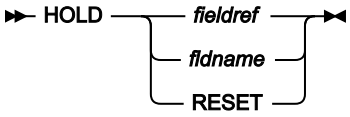
HOLD primary command

In the ZDT/Db2 editor, the HOLD command sets a scrolling lock on one or more columns.

You can hold columns so that, if you scroll left or right in TABL display format, those columns are always displayed on the left of the panel (regardless of how far you scroll).

You can remove scrolling locks with the FREE editor primary command or by using the H prefix command in the template editor.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

Examples

`HOLD RESET`

Removes all scrolling locks.

`HOLD #4`

Sets a scrolling lock on column #4.

`HOLD #9 #2`

Sets a scrolling lock on columns #9 and #2.

Note: The columns are held in the order that is specified, so the display shows (from left to right) columns #9 and #2 followed by the non-held columns.

`HOLD #5-#9`

Sets a scrolling lock on columns #5, #6, #7, #8, and #9.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616

- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

INDEXINF primary command

The INDEXINF primary command shows or hides index information about each column when displayed in SNGL display format.

In SNGL display format, ZDT/Db2 displays an additional column (to the left of the column containing the column name) showing information about primary keys, foreign keys, unique indexes. (If you enter the INDEXINF primary command in TABL display format, the display is unaltered. However, if you change to SNGL display format, the display status of the additional column is reversed.)

If the index information is already displayed, the INDEXINF primary command removes it from display.

Note:

1. Information about primary keys, foreign keys, and unique indexes is only shown for tables (but not for views).
2. Foreign key information can only be displayed if the **Retrieve foreign key information when building templates** option is selected when the template is created.
3. The INDEXINF primary command has the same effect as selecting (or deselecting) the **Show keys, indexes and nullable columns** option.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Viewing and changing Db2 data on page 122

INSRPT primary command

Insert repeat. In advanced SELECT prototyping, if the cursor is positioned on a repeatable item, ZDT/Db2 creates a new repeat instance and selects the element on which the cursor was placed.

Syntax



Availability

Advanced SELECT Prototyping panel on page 451

Related tasks

Using advanced SQL prototyping on page 399

LEFT primary command

- In View or Edit:

  In TABL display format, the LEFT primary command scrolls to the left of your data. (You cannot use LEFT in SNGL display format.)

  The amount (number of panel columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

- When editing a template:

  When you are editing a template, the LEFT primary command:
  - Scrolls the **Row Selection Criteria** field on the Column Selection/Edit panel (the cursor must be positioned on the field at the time you issue the LEFT primary command)
  - Toggles the display (on an 80-character screen) on the Row Selection Criteria panel between the **Column name** and **Data type(length)** details appearing (the cursor must not be positioned on a **Value** field at the time you issue the LEFT primary command)

Syntax

Figure 152. ZDT/Db2 editor session

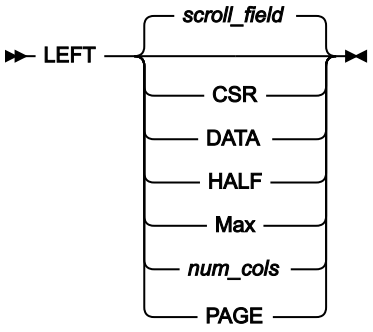


Figure 153. Template editing



scroll_field

Scroll left by the amount indicated in the **Scroll field**. This is the default value if no parameter is used.

CSR

Scroll left to the cursor position.

DATA

Scroll left one panel column less than a page of data.

HALF

Scroll left half a page of data.

Max

Scroll to the beginning of the row.

num_cols

Scroll left *num_cols* panel columns. Valid range: 1–9999.

PAGE

Scroll left one page of data.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Column Selection/Edit panel on page 497
- Row Selection Criteria panel on page 799

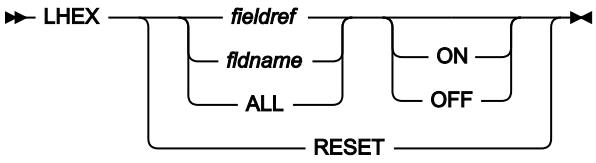
Related tasks

- Scrolling on page 160
- Scrolling the Row Selection Criteria field on page 94
- Changing the display (80-character screens only) on page 91

LHEX primary command

In the ZDT/Db2 editor, the LHEX command displays the data for one or more columns in long hexadecimal format. In LHEX format, each character is shown as two hexadecimal characters, which are displayed from left to right. For example, '1' appears as 'F1' in LHEX format. Only (VAR)CHAR, (VAR)GRAPHIC, DATE, TIME, and TIMESTAMP columns are affected.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

ON|OFF

Indicates whether to turn the LHEX option on or off. Omitting this parameter results in the command acting as a toggle; the field's current LHEX display status is reversed.

Examples

`LHEX ALL ON`

Displays the data for all eligible columns in long hex.

`LHEX #4 ON`

Displays the data for column #4 in long hex.

`LHEX #9 #2 OFF`

Displays the data for columns #9 and #2 in character format.

`LHEX #5-#7 ON`

Displays the data for columns #5, #6, and #7 in long hex.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

LOBBRWS primary command

The LOBBRWS primary command starts a new ZDT/Db2 LOB editor session, in browse mode, for a LOB (BLOB, CLOB or DBCLOB) column. The Db2® object being processed must contain a LOB column.

The command is cursor sensitive. Type the command, move the cursor to the LOB column of interest and press Enter. A Z Data Tools/Db2 LOB editor session, in browse mode, for the LOB column at the cursor position is displayed.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 154. In SNGL display format

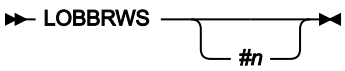
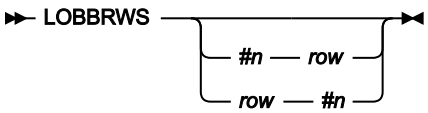


Figure 155. In TABL display format



#n

A valid reference. The data type for *#n* must be BLOB, CLOB or DBCLOB. *#n* can be omitted when the Db2® object contains only one LOB column. If *#n* is omitted when the Db2® object contains multiple LOB columns, ZDT/Db2 searches for the first selected LOB column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

LOBEDIT primary command

The LOBEDIT primary command starts a new ZDT/Db2 LOB editor session, in edit mode, for a LOB (CLOB or DBCLOB) column. Edit of a BLOB is not supported. The Db2® object being processed must contain a LOB column.

The command is cursor sensitive. Type the command, move the cursor to the LOB column of interest and press Enter. A Z Data Tools/Db2 LOB editor session, in edit mode, for the LOB column at the cursor position is displayed.

Note: Do not use the SAVE command in a LOBEDIT session because the LOB data will not be updated in the Db2 table.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 156. In SNGL display format

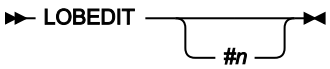
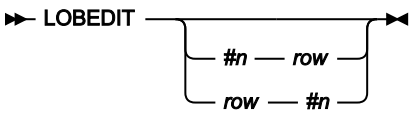


Figure 157. In TABL display format



#n

A valid reference. The data type for *#n* must be a CLOB or DBCLOB. *#n* can be omitted when the Db2® object contains only one LOB column. If *#n* is omitted when the Db2® object contains multiple LOB columns, ZDT/Db2 searches for the first selected LOB column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Edit panel on page 868

LOBLEN primary command

The LOBLEN primary command displays the length for a LOB (BLOB, CLOB or DBCLOB) column. The Db2® object being processed must contain a LOB column.

The command is cursor sensitive. Type the command, move the cursor to the LOB column of interest and press Enter. The length for the LOB column at the cursor position is displayed.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 158. In SNGL display format

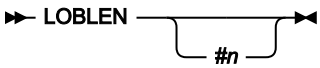
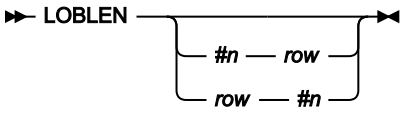


Figure 159. In TABL display format



#n

A valid reference. The data type for *#n* must be BLOB, CLOB or DBCLOB. *#n* can be omitted when the Db2® object contains only one LOB column. If *#n* is omitted when the Db2® object contains multiple LOB columns, ZDT/Db2 searches for the first selected LOB column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

LOBVIEW primary command

The LOBVIEW primary command starts a new ZDT/Db2 LOB editor session, in view mode, for a LOB (BLOB, CLOB or DBCLOB) column. The Db2® object being processed must contain a LOB column

The command is cursor sensitive. Type the command, move the cursor to the LOB column of interest and press Enter. A Z Data Tools/Db2 LOB editor session, in view mode, for the LOB column at the cursor position is displayed.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 160. In SNGL display format

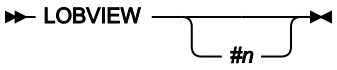
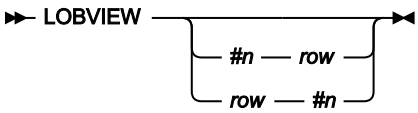


Figure 161. In TABL display format



#n

A valid reference. The data type for *#n* must be BLOB, CLOB or DBCLOB. *#n* can be omitted when the Db2® object contains only one LOB column. If *#n* is omitted when the Db2® object contains multiple LOB columns, ZDT/Db2 searches for the first selected LOB column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

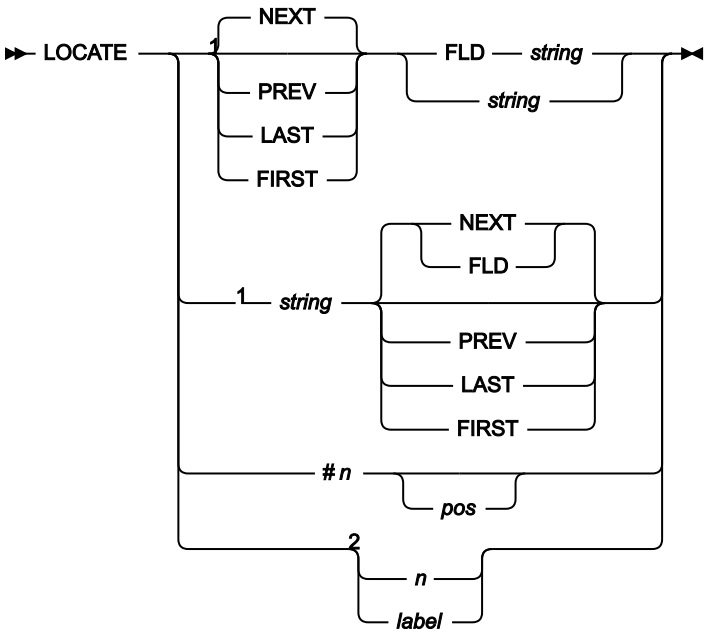
- Table Edit panel on page 868
- Table View panel on page 882

LOCATE primary command

- When you use the LOCATE primary command in an editor session or the Object List utility:
  - ZDT/Db2 moves the display to the row, or first column, that matches the search criteria.
  - To move to a particular row, specify the row number. You can specify a label to move to a row to which that label has previously been assigned.
  - To search for a particular column, specify either the column number (for example, #4), or part or all of a column name (for example, SAL). When you search for part or all of a column name, you can use one of the optional parameters to nominate the direction and starting point for the search. Where there may be ambiguity, use the FLD parameter to differentiate between a column name and a Db2® column number.
- When you use the LOCATE primary command in Print Browse (3.11):
  - Positions a specified line number at the top of the display.

Syntax

Figure 162. ZDT/Db2 editor session



Notes:

[1] These forms of the locate command locate a column.

[2] These forms of the locate command locate a row.

Figure 163. Object List utility

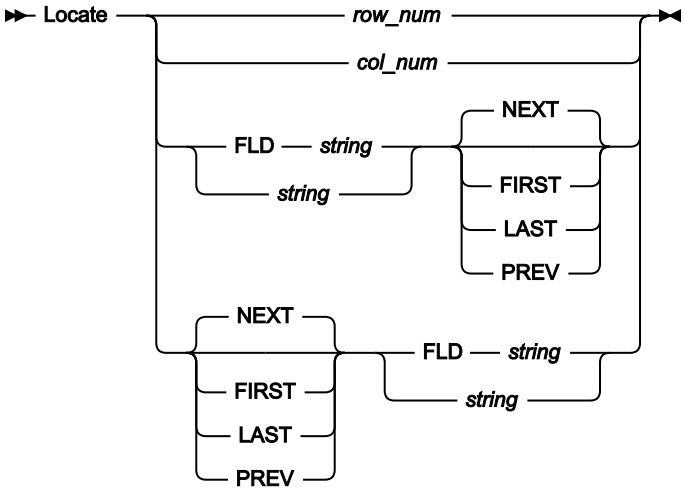


Figure 164. Print Browse



string

A column name, or part of a column name

#n

A valid reference to a column.

pos

An integer representing a character position within a long (>50 characters) character or graphic column.

n

The sequence number of a row.

label

The label of a labeled row. *label* may be a user or system label.

row_num

The number of the row you want to locate. The row number is shown in the prefix area (TABL display mode), or in the upper right of the panel (SNGL display mode). For example, the following command locates row number 23:

```
LOCATE 23
```

Note: The row number is not shown when data is displayed with the Object List utility.

col_num

The Db2® column number (specified as #*n*) of the column you want to locate. For example, the following command locates column #12:

```
L #12
```

label

An existing user-assigned or editor-assigned label, identifying the row that you want to locate. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter *"Z"* indicate an editor-assigned label. For example:

`LOCATE .HERE` locates a row marked with an existing label (.HERE)

`L .ZLST` locates the last row.

FLD

Indicates to ZDT/Db2 that the string (*string*) following the FLD keyword is the name or part of the name of the column you want to locate. For example, the following command locates the column named #ITEMS:

```
L FLD #ITEMS
```

(In this case, if you omit the FLD parameter, the command `L #ITEMS` results in an error as ZDT/Db2 expects the # symbol to be followed by a numeric Db2® column number.)

string

The name (or part of a name) of the column you want to locate. The string can occur anywhere in the column name. For example, both of the following commands locate a column named *"SALARY"*:

```
L SAL
L ARY
```

NEXT

This is the default setting. In TABL display format, the search for *string* starts at (but excludes) the column currently displayed on the left of the screen and scans right. In SNGL display format, the search for *string* starts at (but excludes) the column currently displayed at the top of the screen and scans down. For example, both of the following commands locate the next column whose name contains the string *"SAL"*:

```
L NEXT SAL
L SAL NEXT
```

FIRST

In TABL display format, the search for *string* starts at (but excludes) the first column for the Db2® object and scans right. In SNGL display format, the search for *string* starts at (but excludes) the first column for the Db2® object and scans down. For example, both of the following commands locate the first column whose name contains the string *"SAL"*:

```
L FIRST SAL
L SAL FIRST
```

LAST

In TABL display format, the search for *string* starts at the last column for the Db2® object and scans left. In SNGL display format, the search for *string* starts at the last column for the Db2® object and scans up. For example, both of the following commands locate the last column whose name contains the string *"SAL"*:

```
L LAST SAL
L SAL LAST
```

PREV

In TABL display format, the search for *string* starts at the column currently displayed on the left of the screen and scans left. In SNGL display format, the search for *string* starts at the column currently displayed at the top of the screen and scans up. For example, both of the following commands locate the previous column whose name contains the string *"SAL"*:

```
L PREV SAL
L SAL PREV
```

line_num

The line number you want positioned at the top of the display.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868

- Table View panel on page 882
- Collections panel on page 482
- Columns panel on page 503
- Databases panel on page 609
- Database Request Modules panel on page 607
- Distinct Types panel on page 637
- Functions panel on page 709
- Indexes panel on page 723
- Application Packages panel on page 469
- Application Plans panel on page 473
- Schemas panel on page 812
- Stored Procedures panel on page 852
- Synonyms panel on page 855
- Tables, Views and Aliases panel on page 891
- Table Spaces panel on page 879
- Triggers panel on page 903
- Storage Group(s) panel on page 848

Related tasks

- Locating a row on page 142
- Locating a column on page 142
- Finding a string in a list of Db2 objects on page 314
- Locating a row or column in a list of Db2 objects on page 314

LZERO primary command

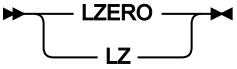
The LZERO primary command toggles the global leading zero setting for the template.

If global leading zeros is set to ON, then the default leading zero setting for applicable numeric fields is YES, and a status indicator **LZERO** appears on the current panel.

If global leading zeros is set to OFF, then the default leading zero setting for applicable numeric fields is NO, and the status indicator is cleared.

For the complete list of software supported by Z Data Tools, see the topic about Supported software environments in the *Z Data Tools Customization Guide*.

Syntax



Availability

- Column Selection/Edit panel on page 497

NEW, NEWS primary command

The NEW (or NEWS) primary command displays a panel providing general information about the current ZDT/Db2 version and release.

Syntax



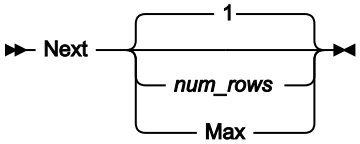
Availability

- All panels

NEXT primary command

Scrolls forward the number of rows specified.

Syntax



num_rows

Scroll forward *num_rows* rows. Valid range: 1–999999999.

Max

Scroll forward to make the last row visible.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Viewing and changing Db2 data on page 122
- Scrolling on page 160

NEXTRPT primary command

In advanced SELECT prototyping, if the cursor is positioned on a repeatable item within the syntax, the NEXTRPT primary command makes the next repeatable item current.

Syntax

Availability

Advanced SELECT Prototyping panel on page 451

Related tasks

Using advanced SQL prototyping on page 399

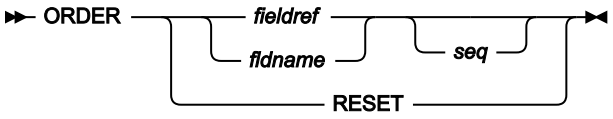
ORDER primary command

In the ZDT/Db2 editor, the ORDER command sets the display order for one or more columns in the TABL and SNGL display format.

Held columns are ordered in relation to other held columns. Unheld columns are ordered in relation to other unheld columns.

You can also specify a display order by typing a number in the SEQ column using the template editor.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

seq

A positive integer that indicates the relative order for the column among other ordered columns.

Examples

`ORDER #9 #2`

Displays columns #9 and #2 on the left.

`ORDER #4 0`

Removes #4 from the list of ordered columns.

`ORDER #6 2`

Displays column #6 as the second ordered column.

`ORDER #5-#7 0`

Removes columns #5 - #7 from the list of ordered columns.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

PB primary command

The PB primary command lets you browse, and optionally purge, the print output from your current ZDT/Db2 editor session held in the print data set.

If you enter the PB primary command when you have not created any print output, ZDT/Db2 displays the message `Empty print data set`.

Syntax



Availability

Available on all panels

Related tasks

- Looking at the print output from your ZDT/Db2 session on page 333

PLAN primary command

When you use the SQL prototyping, execution and analysis option (4), you can use the PLAN primary command:

- On the Plan Table Rows panel to switch panel format to show the plan name.
- On the Statement Table Rows panel to switch panel format to show the plan name.

Syntax



Availability

Plan Table Rows panel on page 754

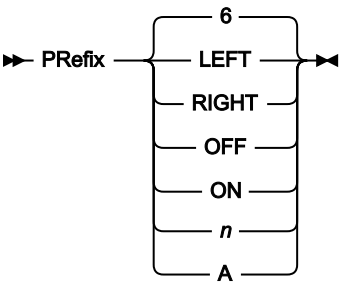
Related tasks

Working with SQL statements on page 390

PREFIX primary command

For TABL display format only, the PREFIX primary command sets the position and display state of the prefix area.

Syntax



LEFT

Displays the prefix area on the left side.

RIGHT

Displays the prefix area on the right side.

OFF

Does not display the prefix area.

ON

Displays the prefix area at the position last set, with the display width last set.

n

Displays the prefix area at the position last set, with a display width of *n* digits. *n* must be in the range: 6–9.

A

Displays the prefix area at the position last set, with a display width of 6 digits or, if necessary, larger (up to 9 digits) in order to display the whole record number.

Availability

- Table Edit panel on page 868

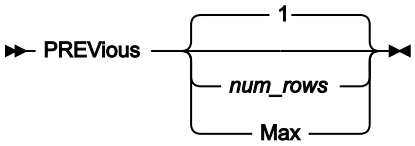
Related tasks

- Managing the prefix area on page 165

PREVIOUS primary command

Scrolls backward the number of rows specified.

Syntax



num_rows

Scroll backward *num_rows* rows. Valid range: 1–999999999.

Max

Scroll backward to make the first row visible.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Viewing and changing Db2 data on page 122
- Scrolling on page 160

PREVRPT primary command

In advanced SELECT prototyping, if the cursor is positioned on a repeatable item within the syntax, the PREVRPT primary command makes the previous repeatable item current.

Syntax

Availability

Advanced SELECT Prototyping panel on page 451

Related tasks

Using advanced SQL prototyping on page 399

PRINT primary command

In Print Browse, the PRINT primary command transfers the contents of the print data set to a SYSOUT class.

You specify the print data set using the **PRINTDSN** entry field on the Set Print Processing Options panel.

The SYSOUT class is determined by one of the following (whichever is found to exist first):

- The class specified by SYSOUT=*c* in the **PRINTOUT** entry field on the Print Utility panel (unless this field contains a value other than SYSOUT=*c*)
- Your default TSO SYSOUT class (if defined)
- SYSOUT class A (if neither of the above apply)

Note: When you enter `PRINT` on the command line for any ZDT/Db2 panel other than the Print Browse panel, it is interpreted as the ISPF system command, PRINT, and records a snapshot of the physical screen image in the print data set.

Syntax



Availability

Any Print Browse display panel

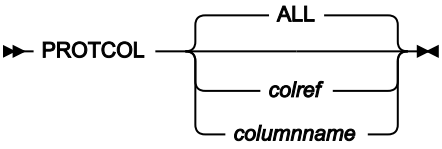
Related tasks

- Looking at the print output from your ZDT/Db2 session on page 333

PROTCOL primary command

In the row selection criteria panel of the Z Data Tools template editor, the PROTCOL command changes the column name and, when displayed, the data type fields from editable to protected. Any column expression entered for a column is retained. You can also enter the PC prefix command against a column to achieve the same result. Use this command to protect a previously entered column expression from inadvertent changes.

Syntax



colref

A reference to a single column, for example, #4.

columnname

The name of a column in the Db2® object or result table being processed.

PROTCOL with no parameters changes all column name fields and, when displayed, data type fields to protected.

PROTCOL with either a column reference or column name changes the specified column name field and, when displayed, data type fields to protected.

Availability

- Row Selection Criteria panel on page 799

Related tasks

- Row selection by column on page 86

PURGE primary command

In Print Browse, the PURGE command purges the print output by clearing the print data set.

Syntax



Availability

A Print Browse display panel

Related tasks

- Looking at the print output from your ZDT/Db2 session on page 333

QUIT primary command

See CANCEL primary command on page 926.

RCHANGE primary command

The RCHANGE primary command changes the next occurrence of the string or numeric value by repeating the previous CHANGE primary command.

Syntax



Availability

- Table Edit panel on page 868

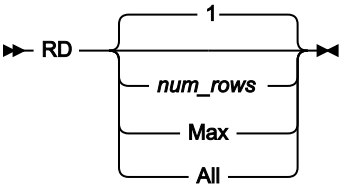
Related tasks

- Changing data with the CHANGE command on page 182

RD primary command

The RD (*"Record Dump"*) primary command prints, for the number of rows specified, in dump format using the current display format (TABL or SNGL).

Syntax



num_rows

Print *num_rows* rows, starting from the current row. Valid range: 1–999999999.

Max

Print all remaining rows, from the current row to the end of the Db2® object.

All

Print all rows in the Db2® object.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Printing rows from a Db2 table on page 331

REDIT primary command

When you issue the REDIT primary command from within an information panel for a referential integrity error, ZDT/Db2 displays an Edit session.

When you issue the REDIT primary command from within a Z Data Tools/Db2 View or Edit editor session, ZDT/Db2 displays the Related tables panel listing information about the tables related to the table you are currently editing.

Note:

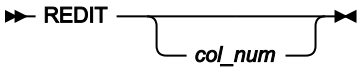
1. The REDIT primary command is not available in ZDT/Db2 Browse.
2. The REDIT command works with Db2® tables only. You cannot use the command with Db2® views.

Syntax

Figure 165. Syntax (when issued for a referential integrity error)



Figure 166. Syntax (when issued from within a Z Data Tools/Db2 editor session)



col_num

A Db2® column number (specified as *#n*).

Availability

- Within an information panel for a referential integrity error.
- From within a Z Data Tools/Db2 editor session.

Related tasks

- Starting an Edit session of a related table on page 195
- Listing related tables on page 193

REFRESH primary command

The REFRESH primary command refreshes the list of Db2® subsystems on the Db2® Subsystem Selection menu.

Syntax



Availability

Db2 Subsystem Selection panel on page 624

Related tasks

Selecting the Db2 subsystem when more than one is available on page 30

REFS primary command

The REFS primary command shows or hides the column number (as defined in the Db2® catalog) for each displayed column when in SNGL display format.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Displaying the column number on page 167

RESET primary command

In a Z Data Tools/Db2 editor session, you use the RESET command to redisplay excluded rows, or to reset strings highlighted by the FIND command. You can also use the RESET command to reset pending prefix commands, or to remove labels from rows.

For basic SELECT prototyping, the RESET primary command clears previous information from the SELECT, WHERE, and ORDER BY clauses in the SELECT statement at the top of the Basic SELECT prototyping panel.

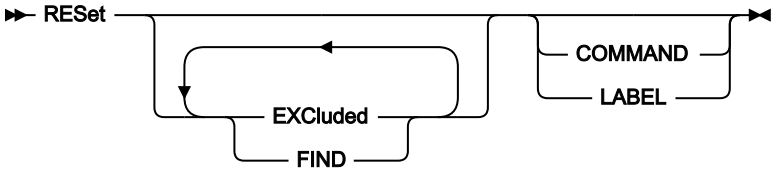
When you are viewing the Db2® utility options panels, the RESET primary command sets the values for each option back to the installation default.

Syntax

Figure 167. Standard



Figure 168. ZDT/Db2 editor session



COMMAND

All pending prefix commands are reset.

EXCLuded

All excluded rows are redisplayed.

FIND

All strings highlighted by the FIND command are reset.

LABEL

All row labels are deleted.

Availability

- Table View panel on page 882
- Table Browse panel on page 866
- Table Edit panel on page 868
- Basic SELECT Prototyping panel on page 476

Related tasks

- Viewing and changing Db2 data on page 122
- Using basic SQL prototyping on page 390
- Setting options for Db2 utilities on page 348

RFIND primary command

The RFIND primary command repeats the search performed by the previous FIND primary command.

Syntax

RFind

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Repeating the search on page 146

RIGHT primary command

- In View or Edit:

  In TABL display format, the RIGHT primary command scrolls to the right of your data. (You cannot use RIGHT in SNGL display format.)

  The amount (number of panel columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

- When editing a template:

  When you are editing a template, the RIGHT primary command:
  - Scrolls the **Row Selection Criteria** field on the Column Selection/Edit panel (the cursor must be positioned on the field at the time you issue the RIGHT primary command)
  - Toggles the display (on an 80-character screen) on the Row Selection Criteria panel between the **Column name** and **Data type(length)** details appearing (the cursor must not be positioned on a **Value** field at the time you issue the RIGHT primary command)

Syntax

Figure 169. View or Edit

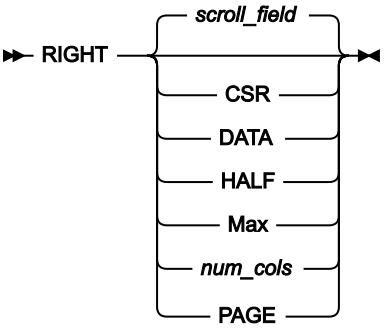


Figure 170. Template editing



scroll_field

Scroll right by the amount indicated in the **Scroll field**. This is the default value if no parameter is used.

CSR

Scroll right to the cursor position.

DATA

Scroll right one panel column less than a page of data.

HALF

Scroll right half a page of data.

Max

Scroll to the end of the row.

num_cols

Scroll right *num_cols* panel columns. Valid range: 1–9999.

PAGE

Scroll right one page of data.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Column Selection/Edit panel on page 497
- Row Selection Criteria panel on page 799

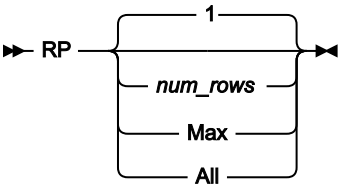
Related tasks

- Scrolling on page 160
- Scrolling the Row Selection Criteria field on page 94
- Changing the display (80-character screens only) on page 91

RP primary command

The RP (*"Record Print"*) primary command prints, for the number of rows specified, in character format using the current display format (TABL or SNGL).

Syntax



num_rows

Print *num_rows* rows, starting from the current row. Valid range: 1–999999999.

Max

Print all remaining rows, from the current row to the end of the Db2® object.

All

Print all rows in the Db2® object.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Printing rows from a Db2 table on page 331

RUNTEMP primary command

The RUNTEMP primary command runs the current function with the template changes you have made but does not save the changes. RUNTEMP is only available when you run a function that uses a template.

Syntax



Availability

- Template Save panel on page 902
- Column Selection/Edit panel on page 497

Related tasks

- Viewing and changing Db2 data on page 122
- Printing the contents of a Db2 table on page 328
- Copying data from one Db2 object to another on page 246
- Copying data from a VSAM or QSAM file on page 251
- Copying data to a VSAM or QSAM file on page 279
- Populating a Db2 table with data on page 244
- LOAD utility on page 357

SAVE primary command

In Edit, the SAVE primary command saves any uncommitted changes without ending the Edit session.

Syntax



Availability

- Table Edit panel on page 868

Related tasks

- Ending a Z Data Tools/Db2 editor session on page 133
- Updating a primary key on page 199

SAVEAS primary command

When you edit a template, the SAVEAS primary command displays the save panel where you can save the template as another name.

Syntax



Availability

- Column Selection/Edit panel on page 497

Related tasks

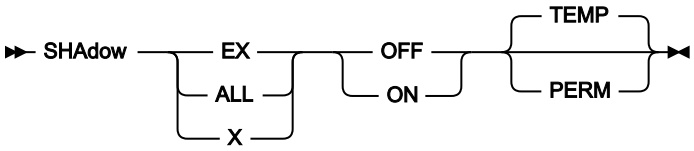
- Working with templates on page 74

SHADOW primary command

The SHADOW command hides or shows shadow lines.

Note: Shadow lines represent rows that have been excluded from display by the EXCLUDE primary command.

Syntax



EX

Show or hide "excluded" rows shadow lines. This is the default setting.

ALL

Show or hide "excluded" rows shadow lines.

X

Same as EX.

OFF

Hide "excluded" rows shadow lines.

ON

Show "excluded" rows shadow lines.

TEMP

Shadow setting applies to the current ZDT/Db2 editor session only. This is the default setting.

PERM

Shadow setting is saved in your ZDT/Db2 profile, and applies to all ZDT/Db2 editor sessions until changed.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

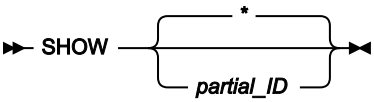
Related tasks

- Excluding rows on page 162

SHOW primary command

The SHOW command restricts the list of Db2® subsystems shown on the Db2® Subsystem Selection menu.

Syntax



*****

Show all Db2® subsystems.

partial_ID

The partial ID of a Db2® subsystem, starting or ending with an asterisk (*), or both. Shows all Db2® subsystems whose ID matches the partial ID specified.

Availability

- Db2 Subsystem Selection panel on page 624

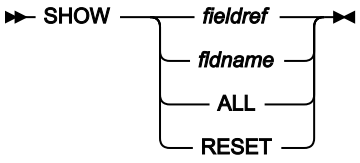
Related tasks

- Restricting the list of Db2 subsystems on page 34

SHOW primary command

In the ZDT/Db2 editor, the SHOW command adds a hidden column to the data display. You can hide a column from display by using the HIDE editor primary command or by using the S prefix command in the template editor.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

Examples

`SHOW ALL`

Shows all columns.

`SHOW RESET`

Shows all columns.

`SHOW #4`

Shows the data for column #4.

`SHOW #7 #1`

Shows the data for columns #1 and #7.

`SHOW #2-#5`

Shows the data for columns #2, #3, #4, and #5.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

SHOWCOB primary command

The SHOWCOB command displays details showing which COBOL compiler is currently being used.

Syntax



Availability

Available on all panels

Related tasks

- Checking which COBOL compiler you're using on page 29

SHOWPLI primary command

The SHOWPLI command displays details showing which PL/I compiler is currently being used.

Syntax



Availability

Available on all panels

Related tasks and examples

- Checking which PL/I compiler you are using on page 29

SORT primary command

- In View or Edit, the SORT primary command changes the order of the data you are browsing or editing in TABL display format.

  ZDT/Db2 sorts the data according to its hexadecimal representation using a hierarchy of keys, starting with the first specified column as the primary key, and subsequent specified columns as less significant keys. You specify the hierarchy left to right.
- When you use the Object List utility (3.4) to display a list of objects, you can sort a column in the current object list by issuing the SORT primary command with the cursor positioned on the column you want to sort.

Syntax

Figure 171. Syntax (View or Edit)

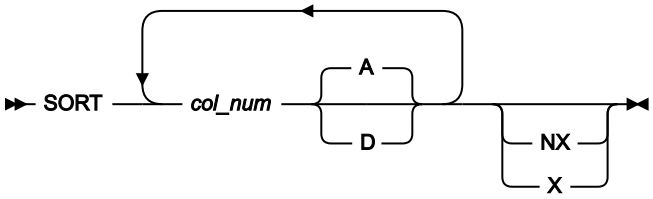


Figure 172. Syntax (Object List utility)



col_num

The Db2® column number (specified as *#n*) of a column to be used as the sort key. You can specify between 1 and 5 column numbers. Multiple column numbers must be separated by a blank.

A

Sort data in ascending sequence. This is the default sequence.

D

Sort data in descending sequence.

NX

Non-excluded rows only. Sorts rows as if any excluded rows did not exist.

X

Excluded rows only. Sorts excluded rows as if any non-excluded rows did not exist.

Availability

- Db2 View panel on page 631
- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

SQL primary command

The SQL primary command displays the SQL SELECT statement for the object you are processing. You can view the SELECT statement and save it in an ISPF data set for future reference.

When you enter the SQL command, the SELECT statement is presented in a standard ISPF EDIT panel. You can save the statement using the ISPF CREATE command. To do this:

1. On the command line, type `CREATE`.
2. In the prefix area of the top line of the SELECT statement, type `C9999`.
3. Press Enter.

   You are asked to supply the data set and member name in which to save the SELECT statement. If the data set does not exist, you are asked to specify the allocation attributes to create it. You cannot use the attributes of the temporary data set displayed; you must provide the required attributes yourself.

Note: When you start an editor session from the function entry panel, the SELECT clause in the SQL statement includes every column of the table, even if you have edited the template and deselected one or more columns.

When you are working with SQL statements, any changes you make to the command in the ISPF Edit session are not preserved when you return to the prototyper.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Basic SELECT Prototyping panel on page 476

- Advanced SELECT Prototyping panel on page 451
- Enter, Execute and Explain SQL Statements panel on page 684

Related tasks

- Viewing and changing Db2 data on page 122
- Working with SQL statements on page 390

SQLID primary command

The SQLID primary command allows you to change the SQLID to be used as part of the connection to Db2®.

Syntax



DB2_sqlid

The required SQLID to be used as part of the connection to Db2®.

Availability

- The SQLID command can be issued on most ZDT/Db2 processing panels.

  You cannot issue the command from:

  - Pop-up panels
  - Tutorial panels
  - Panels displayed by ISPF. For example, the Audit log listing.

Related tasks

- Specifying the SQLID on page 52

TEDIT primary command

The TEDIT command displays the Column Selection/Edit panel from within a Z Data Tools/Db2 editor session.

Syntax



Availability

- Table Edit panel on page 868
- Table Browse panel on page 866
- Table View panel on page 882

Related tasks

- Editing a template on page 84

TOP primary command

The TOP primary command scrolls to the first page of data.

Syntax



Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Viewing and changing Db2 data on page 122
- Scrolling on page 160

TP primary command

The TP primary command prints the current template.

Syntax



Availability

- Column Selection/Edit panel on page 497

Related tasks

- Printing a template on page 77

TYPE primary command

The TYPE primary command shows or hides the data type (and, for non-numeric columns, the length of the column) for each displayed column when in SNGL display format.

Syntax

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Displaying the data type on page 167

UNDO primary command

Use the UNDO primary command to undo the last set of changes you made to the SELECT statement at the top of the Basic SELECT prototyping panel.

Syntax

Availability

Basic SELECT Prototyping panel on page 476

Related tasks

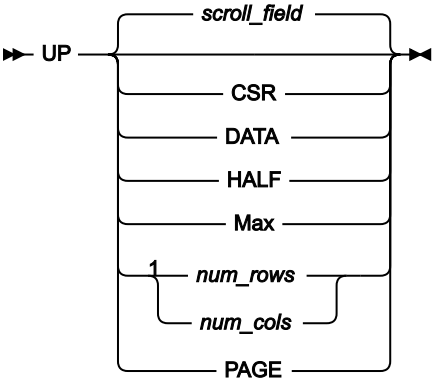
Using basic SQL prototyping on page 390

UP primary command

The UP primary command scrolls up (backward) through your data.

The amount (number of rows or columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax



Notes:

[1] *num_rows* available in TABL display format; *num_cols* available in SNGL display format.

scroll_field

Scroll up by the amount indicated in the **Scroll field**. This is the default value if no parameter is used.

CSR

Scroll up to the cursor position.

DATA

Scroll up one row (TABL) or column (SNGL) less than a page of data.

HALF

Scroll up half a page of data.

Max

Scroll to top of the table. This has the same effect as TOP command.

num_rows

Scroll up *num_rows* rows. Valid range: 1–9999. Available in TABL display format.

num_cols

Scroll up *num_cols* columns. Valid range: 1–9999. Available in SNGL display format.

PAGE

Scroll up one page of data.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882
- Enter, Execute and Explain SQL Statements panel on page 684

Related tasks

- Viewing and changing Db2 data on page 122
- Scrolling on page 160
- Entering, executing, and explaining SQL statements on page 424

VER primary command

The VER primary command displays the current ZDT/Db2 version number and the PTF number for each ZDT/Db2 component in a window. The window also indicates whether or not ZDT/Db2 is APF-authorized.

Syntax



Availability

Available on all panels

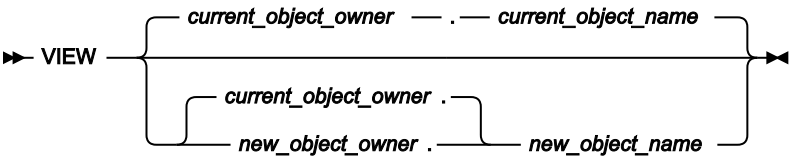
Related tasks

Checking your ZDT/Db2 version on page 27

VIEW primary command

The VIEW primary command starts a new ZDT/Db2 View editor session.

Syntax



current_object_owner

The name of the owner of the object in the current ZDT/Db2 editor session.

current_object_name

The name of the object in the current ZDT/Db2 editor session.

new_object_owner

The name of the owner of the object to be pre-filled in the entry panel for the new ZDT/Db2 editor session.

new_object_name

The name of the object to be pre-filled in the entry panel for the new ZDT/Db2 editor session.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

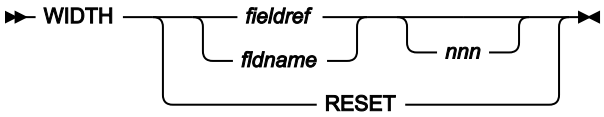
Related tasks

- Viewing and changing Db2 data on page 122

WIDTH primary command

The WIDTH command sets the display width for one or more columns in TABL display format. You can also specify a display width by specifying a value in the output width field on the column attributes panel in the template editor; use the E prefix command to access the column attributes panel.

Syntax



fieldref

A single field reference (for example, `#1`), a range (for example, `#2-#5`), or a list of single field references/ranges separated by spaces (for example, `#1 #4 #6-#7`). A single field specification is `#n`; n is a positive integer and cannot exceed the number of columns in the Db2® object or result table being processed.

fldname

The name of a column in the Db2® object or result table being processed. The column name can be preceded by a pound sign (#). Any value entered is translated to upper case if the "Translate Db2® object names" system's option is selected.

nnn

A positive integer that represents the desired display width for the fields. If this parameter is omitted, the display width is reset to the default value. The minimum value of *nnn* is 6. The maximum value of *nnn* depends on the data type:

Table 26. Maximum value of parameter *nnn* for different data types

Data Type	Maximum value of *nnn*
DATE, TIME, or BINARY	30
binary floating point	35
TIMESTAMP	42
DECIMAL	45
decimal floating point	52
ZONED TIMESTAMP	165
Other data types	If the value of the display length is less than 20, the maximum value of *nnn* is 30. Otherwise, the maximum value of *nnn* is the value of the display length plus 10.

Examples

```
WIDTH #9 30
```

Sets the display width for column 9 to 30 bytes.

```
WIDTH #5-#7 20
```

Sets the display width for columns #5 - #7 to 20 bytes.

Availability

- Db2 Browse panel on page 612
- Db2 Edit panel on page 616
- Db2 View panel on page 631
- Any panels displayed with the Object List utility (3.4)

Related tasks

- Viewing and changing Db2 data on page 122
- Working with lists of Db2 objects on page 303

XMLBRWS primary command

The XMLBRWS primary command starts a new ISPF editor session, in browse mode, for an XML document. The Db2® object being processed must contain an XML column.

The command is cursor sensitive. Type the command, move the cursor to the XML column of interest and press Enter. An ISPF editor session, in browse mode, for the XML column at the cursor position is displayed.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 173. In SNGL display format

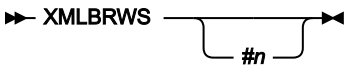
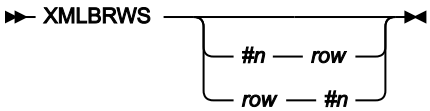


Figure 174. In TABL display format



#n

A valid reference. The data type for *#n* must be XML. *#n* can be omitted when the Db2® object contains only one XML column. If *#n* is omitted when the Db2® object contains multiple XML columns, ZDT/Db2 searches for the first selected XML column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

XMLEDIT primary command

The XMLEDIT primary command starts a new ISPF editor session, in edit mode, for an XML document. The Db2® object being processed must contain an XML column.

The command is cursor sensitive. Type the command, move the cursor to the XML column of interest and press Enter. An ISPF editor session, in edit mode, for the XML column at the cursor position is displayed.

Note: Do not use the SAVE command in am XMLEDIT session because the XML data will not be updated in the Db2 table.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 175. In SNGL display format

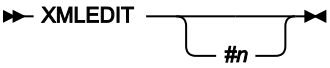
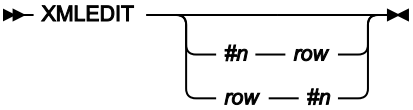


Figure 176. In TABL display format



#n

A valid reference. The data type for *#n* must be XML. *#n* can be omitted when the Db2® object contains only one XML column. If *#n* is omitted when the Db2® object contains multiple XML columns, ZDT/Db2 searches for the first selected XML column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Edit panel on page 868

XMLLEN primary command

The XMLLEN primary command displays the length for an XML document. The Db2® object being processed must contain an XML column.

The command is cursor sensitive. Type the command, move the cursor to the XML column of interest and press Enter. The length for the XML document at the cursor position is displayed.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 177. In SNGL display format

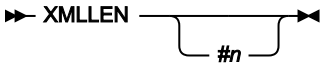
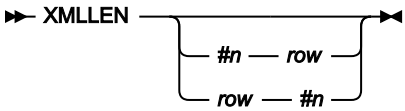


Figure 178. In TABL display format



#n

A valid reference. The data type for *#n* must be XML. *#n* can be omitted when the Db2® object contains only one XML column. If *#n* is omitted when the Db2® object contains multiple XML columns, ZDT/Db2 searches for the first selected XML column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

XMLVIEW primary command

The XMLVIEW primary command starts a new ISPF editor session, in view mode, for an XML document. The Db2® object being processed must contain an XML column.

The command is cursor sensitive. Type the command, move the cursor to the XML column of interest and press Enter. An ISPF editor session, in view mode, for the XML column at the cursor position is displayed.

The command can also be issued without cursor positioning, in which case parameters may be required:

Syntax

Figure 179. In SNGL display format

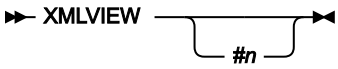
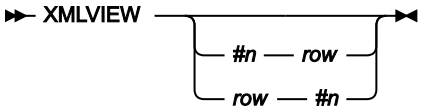


Figure 180. In TABL display format



#n

A valid reference. The data type for *#n* must be XML. *#n* can be omitted when the Db2® object contains only one XML column. If *#n* is omitted when the Db2® object contains multiple XML columns, ZDT/Db2 searches for the first selected XML column, starting at #1.

row

A number that identifies a row in the editor session. A row number should not be specified when the command is issued in SNGL mode, the command applies to the currently displayed row.

Availability

- Table Edit panel on page 868
- Table View panel on page 882

ZOOM primary command

In TABL display format, the ZOOM command zooms in on the row selected by the cursor, or the top row displayed if the cursor is not positioned on a row.

In SNGL display format, the ZOOM command shows the current row. If some columns have been deselected in the template, these columns are now shown.

Syntax

Availability

- Table Browse panel on page 866
- Table Edit panel on page 868
- Table View panel on page 882

Related tasks

- Zooming in to see all of a row on page 158

Chapter 19. ZDT/Db2 functions

This chapter contains information about the ZDT/Db2 functions that can be executed in batch jobs.

The preferred method for generating the ZDT/Db2 batch utility control statements is to allow ZDT/Db2 to generate the required statements.

The syntax diagrams and keyword descriptions in this section are provided for reference only.

How to use this reference chapter

This chapter lists the ZDT/Db2 batch functions in alphabetical order.

The following subsections are supplied, where relevant, for each function:

Purpose

A short summary of what you can do with the function.

Usage notes

A more detailed explanation of how to use the function.

Syntax

A syntax diagram, followed by a parameter list. The parameter list describes the parameter, and shows (where appropriate) its default and other possible values. If you specify parameters that are not relevant to the function, Z Data Tools ignores them. If you specify the same parameter more than once, Z Data Tools uses the first value that you specify.

Batch example

A listing of a sample batch job.

Return codes

Lists the return codes that are specific to the function.

Related functions

Other functions that are related in some way (for example, the export and import functions can be used to transfer data between Db2® systems)

General tips about performance when you use Z Data Tools functions

The following tips are provided as a guide to improving performance when using Z Data Tools functions:

Use the FASTREXX subset of REXX

If a user PROC is required, it is strongly recommended that, rather than the full REXX language, the FASTREXX subset of REXX is used. Z Data Tools provides a set of external REXX functions designed to allow many common tasks to be performed within the FASTREXX subset. See Z Data Tools User’s Guide and Reference Chapter 14, "Enhancing Z Data Tools processing" for a discussion of FASTREXX.

Template and copybook processing

The utilities DBX and DBI allow a copybook or template to be specified for the target/input data set, respectively. The utilities DBX, DBI, DBC, D2G, and DBP optionally require a Db2® template that describes the Db2® object to be processed. When using these functions, consider the following performance-related points:

- Z Data Tools can be most efficient if a template is used instead of a copybook. If a copybook must be used, specify the language of the copybook. (LANG=COBOL, LANG=HLASM, or LANG=PLI, instead of the default LANG=AUTO) for best performance.
- Template expressions should be written, if at all possible, using expressions suitable for Z Data Tools internal processing. For more details, see Z Data Tools User's Guide and Reference "Improving performance using internal expression processing".

ZDT/Db2 batch functions

This section describes the ZDT/Db2 functions that you can use in batch jobs. For more information about using these functions, see Populating a Db2 table with data on page 244, Copying data on page 246, and Printing on page 328.

BATSQL (Batch SQL) batch command

Purpose

To run an SQL statement (delete, insert, or update) that might be issued directly. The number of rows that might be changed by the statement is so large that lock escalation on the target Db2® table is an issue. To avoid lock escalation problems, a three-stage process is used to run the statement:

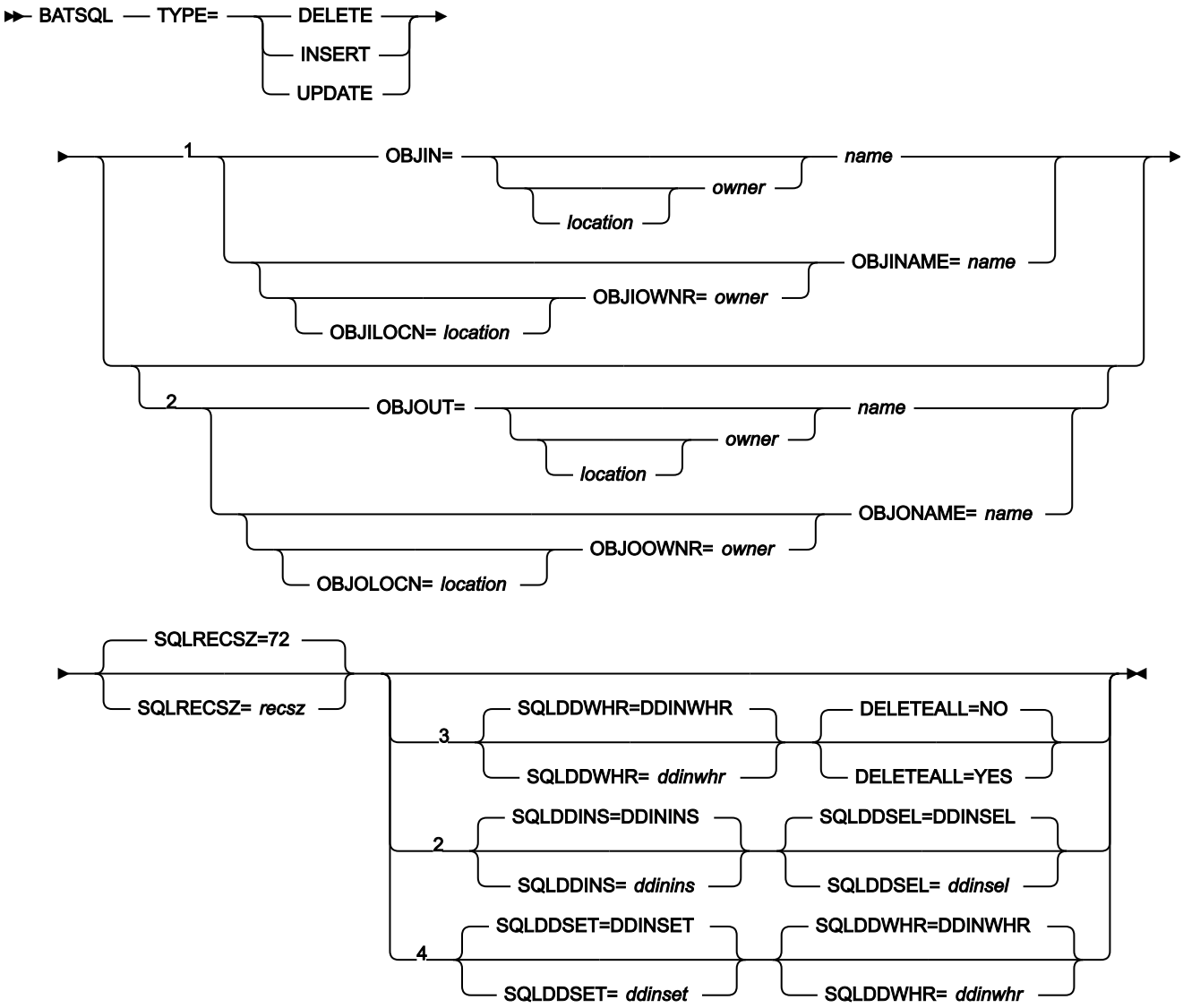
1. An SQL select statement is generated. The result table for this statement includes all of the rows that are subject to delete, insert, or update.
2. Rows from the result table are retrieved and the requested operation (delete, insert, or update) is applied to each row. Rows can be retrieved either one-at-a-time, or in a rowset (more than one row is retrieved at once).
3. After the stipulated number of rows are processed, a Db2® commit is issued to release any locks currently held.

Usage

The ZDT/Db2 batch SQL utility can take these actions:

- Delete rows from a specified table. Potentially all rows might be deleted if no WHERE clause is specified.
- Update rows in a specified table. Potentially all rows might be updated if no WHERE clause is specified.
- Insert rows into a specified table, by using the result table from a full select to generate the rows to be inserted.
- Issue Db2® commits after a specified number of changes

The BATSQL is a batch-only function, there is no equivalent full-screen option. The BATSQL function runs a **single** SQL DELETE, INSERT, or UPDATE operation.



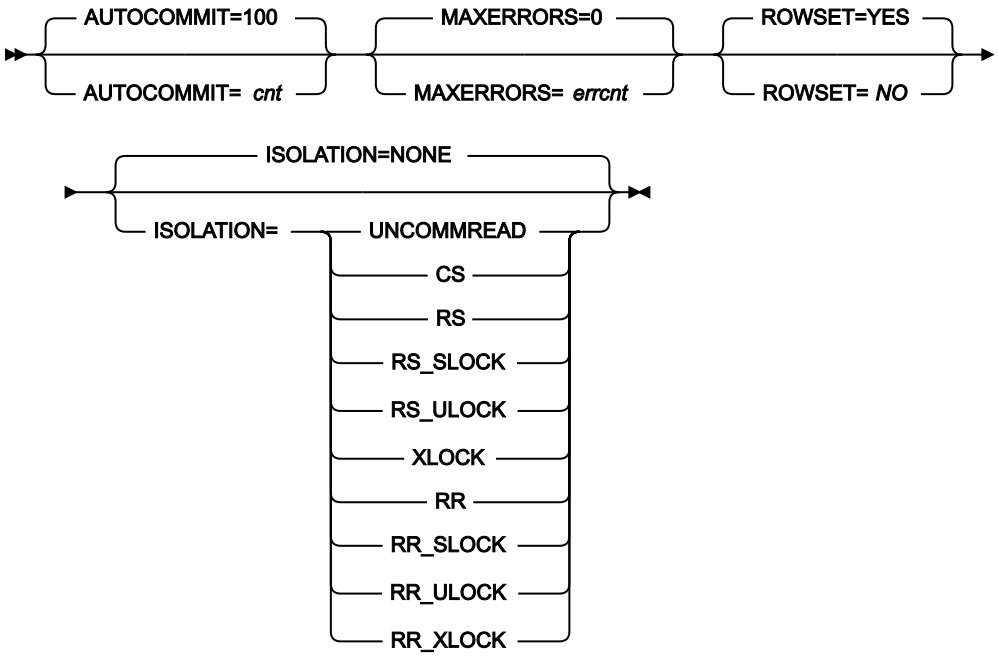
Notes:

[1] TYPE=DELETE, UPDATE

[2] TYPE=INSERT

[3] TYPE=DELETE

[4] TYPE=UPDATE



AUTOCOMMIT=*cnt*

The number of successful operations that are processed before an automatic Db2® commit is issued. A value of zero indicates that no Db2® commits are issued until all rows are processed. Valid values are 0 - 2147483647.

The Db2® cursor that is used to read rows from the result table is declared 'WITH HOLD', so the Db2® commit does not close the cursor. Any locks that are associated with change activities are removed by the Db2® commit.

ISOLATION=*value*

An optional isolation specification to be included when the cursor that accesses the result table is declared. The default is NONE.

See the description of 'isolation clause' in the SQL Reference manual, 'SELECT' statement for a description of the values that can be specified.

UNCOMMREAD (UR) can be specified for TYPE=DELETE, UPDATE but is ignored and converted to CS.

MAXERRORS=*errcnt*

The maximum number of errors that can occur before a Db2® rollback is issued and function processing is terminated. When a row is processed, any negative SQLCODE is considered to be an error, and the error count is incremented by 1. When a rowset is processed, any negative SQLCODE is considered to be an error, and the error count is incremented by the number of rows in the current rowset. Valid values are 0 - 2147483647.

DELETEALL=<YES|NO>

A specification for delete processing, indicating that the intention is to delete all rows from the table. For TYPE=DELETE processing without a where clause, you must specify DELETEALL=YES.

This option is a safety feature, since specifying a delete without a where clause attempts to delete all rows from the table.

OBJIN=*location.owner.name*

Specifies three items:

- The optional name of the Db2® remote server (*location*) where the object is located.
- The optional name of the owner of the object (*owner*).
- The object name (*name*) to be processed.

When location is not specified, the current (local) Db2® server is used. When the owner is not specified, the object name is qualified with the current SQLID.

Use OBJIN when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJILOCN, OBJIOWNR, OBJINAME keywords to specify the object.

OBJILOCN=*location*

The optional name of the Db2® remote server (*location*) where the object is located. See Specifying a Db2 object name on page 1063.

OBJIOWNR=*owner*

The optional name of the owner of the object (*owner*) to be processed. See Specifying a Db2 object name on page 1063.

OBJINAME=*name*

The object name (*name*) for the object to be processed. See Specifying a Db2 object name on page 1063.

Note: Specify OBJIN, OBJILOCN, OBJIOWNR, and OBJINAME for TYPE=DELETE and TYPE=UPDATE.

ROWSET=<YES|NO>

A specification that indicates whether ZDT/Db2 should use rowset processing when it fetches rows, and possibly when it deletes and updates rows.

YES

Rowset processing should be used. The number of rows in each rowset is the same as the value specified for AUTOCOMMIT. If the AUTOCOMMIT keyword is omitted, the rowset size is the default (100 rows). Valid rowset sizes are 2 - 10000. When the AUTOCOMMIT value is outside this range, single-row processing is automatically selected.

NO

Use single-row processing.

OBJOUT=*location.owner.name*

Specifies three items:

- The optional name of the Db2® remote server (*location*) where the target object is located.
- The optional name of the owner of the target object (*owner*)
- The target object name (*name*) to be processed.

When location is not specified, the current (local) Db2® server is used. When the owner is not specified, the target object name is qualified with the current SQLID.

Use OBJOUT when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJILOCN, OBJIOWNR, OBJINAME keywords to specify the target object.

OBJOLOCN=*location*

The optional name of the Db2® remote server (*location*) where the target object is located. See Specifying a Db2 object name on page 1063.

OBJOOWNR=*owner*

The optional name of the owner of the target object (*owner*) to be processed. See Specifying a Db2 object name on page 1063.

OBJONAME=*name*

The object name (*name*) for the target object to be processed. See Specifying a Db2 object name on page 1063.

Note: Specify OBJOUT, OBJOLOCN, OBJOOWNR, and OBJONAME for TYPE=INSERT.

SQLDDWHR=*ddinwhr* (TYPE=DELETE, UPDATE)

The optional DD name that refers to a valid SQL where clause.

If this keyword is omitted, the where clause is assumed to be specified with DD DDINWHR.

For TYPE=DELETE, you can specify a null where clause, however you must also include the keyword DELETEALL=YES to process the delete statement.

SQLDDSET=*ddinset* (TYPE=UPDATE)

The optional DD name that refers to a valid SQL set clause (for TYPE=UPDATE only).

If this keyword is omitted, the set clause is assumed to be specified with DD DDINSET.

A SET clause must be specified.

SQLDDINS=*ddinins* (TYPE=INSERT)

The optional DD name that refers to a valid column list appropriate for an SQL insert statement, for example:

```
INSERT INTO <object>  (column name1, column name2, ..)
```

If this keyword is omitted, the column list is assumed to be specified by using DD DDININS.

A list of column names is optional. If a list of column names is not specified, the result table from the select statement (specified by using SQLDDSEL) ought to have the same number of columns as the number of columns in the target table. Every column name in the list must be a column name of the target table.

SQLDDSEL=*ddinsel* (TYPE=INSERT)

The optional DD name that refers to a select statement that specifies the result table for the rows to be inserted into the target object.

If this keyword is omitted, the select statement is assumed to be specified with DD DDINSEL. A select statement that identifies the rows to be inserted, is required.

SQLRECSZ=*recsz*

The optional record size for all input data sets. The default is 72. When the SQL clauses are specified in-line in the JCL deck, do not specify a value greater than 72. Valid values are 40 - 32760.

Example

Hints and tips

Rowset processing is automatically selected when an AUTOCOMMIT value of 2 - 10000 is specified.

For DELETE, INSERT and UPDATE processing, ZDT/Db2 fetches the rowset number of rows in a single fetch. For DELETE and UPDATE processing only, ZDT/Db2 attempts to delete or update rowset number of rows by using a single SQL statement.

For insert, each row is always inserted by using a single-row INSERT statement.

A large region size can be required when using rowset processing and a large rowset size. ZDT/Db2 needs sufficient storage to store the data for the specified rowset number of rows.

For delete and update processing that uses rowsets, it is possible that the delete or update operation for the rowset fails. When this failure occurs ZDT/Db2 tries the delete or update operation again, for every row of the rowset, using single-row processing. The error that is encountered for the rowset delete or update does not increment the error counter, however any error against a single-row delete or update does increment the error counter.

Example

Examples

The examples refer to these table definitions:

```
CREATE TABLE RFM0681A."Names"
```

```
  ("Name_Identifier" INTEGER NOT NULL
  ,"Last_Name"       VARCHAR(16) NOT NULL
  ,"First_Name"      VARCHAR(16) NOT NULL
  ,"Second_Name"     VARCHAR(16) NOT NULL
  ,"Sex"             CHAR(1)     NOT NULL
  ,"Date_Of_Birth"   DATE        NOT NULL
  );

CREATE TABLE RFM0681B."Names"
  ("Name_Identifier" INTEGER NOT NULL
  ,"Date_Of_Birth"   DATE
  ,"Sex"             CHAR(1)
  ,"First_Name"      VARCHAR(16)
  ,"Second_Name"     VARCHAR(16)
  ,"Last_Name"       VARCHAR(16)
  );
```

Example 1

Delete all rows from the table that do not have a valid sex ('F' or 'M'). Issue a Db2® commit after every 50 successful delete operations. Stop if there are any errors.

```
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DFXX,SQID=XXXXXXX')
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//DDINWHR  DD *
WHERE ("Sex" <> 'M'
  AND  "Sex" <> 'F')
/*
//SYSIN DD *
$$FILEM BATSQL TYPE=DELETE,
$$FILEM OBJIN=RFM0681A."Names",
$$FILEM AUTOCOMMIT=50,
$$FILEM MAXERRORS=1
```

Example 2

Update all rows from the table that have:

- A surname of 'Smith'
- A sex of 'm'

Change these rows to have:

- A surname of 'SMITH'
- A given name of 'JAMES'
- No second name
- A sex of 'M'

Issue a Db2® commit after every 100 successful update operations. Stop if there are more than 10 errors.

```
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DFXX,SQID=XXXXXXX')
```

```
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//DDINSET  DD *
SET ("Last_Name", "First_Name", "Second_Name", "Sex") =
    ('SMITH',     'JAMES',      '',            'M')
/*
//DDINWHR  DD *
WHERE "Last_Name" = 'Smith'
  AND "Sex" = 'm'
/*
//SYSIN DD *
$$FILEM BATSQL TYPE=UPDATE,
$$FILEM OBJIN=RFM0681A."Names",
$$FILEM AUTOCOMMIT=100,
$$FILEM MAXERRORS=10
/*
```

Example 3

Insert rows into RFM0681B."Names", using a select against RFM0681A."Names" to produce a result table.

Only insert rows when the "Name_Identifier" value is less than or equal to 100,000.

```
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DFXX,SQID=XXXXXXX')
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//DDINS  DD *
(
 "First_Name",
 "Second_Name",
 "Last_Name",
 "Sex",
 "Date_Of_Birth",
 "Name_Identifier"
)
/*
//DDINSEL DD *
SELECT
 "First_Name",
 "Second_Name",
 "Last_Name",
 "Sex",
 "Date_Of_Birth",
 "Name_Identifier"
FROM RFM0681A."Names"
WHERE "Name_Identifier" <= 100000
/*
//SYSIN DD *
$$FILEM BATSQL TYPE=INSERT,
$$FILEM OBJOUT=RFM0681B."Names",
$$FILEM AUTOCOMMIT=50,
$$FILEM MAXERRORS=100
/*
```

DBC (Copy) batch command

Purpose

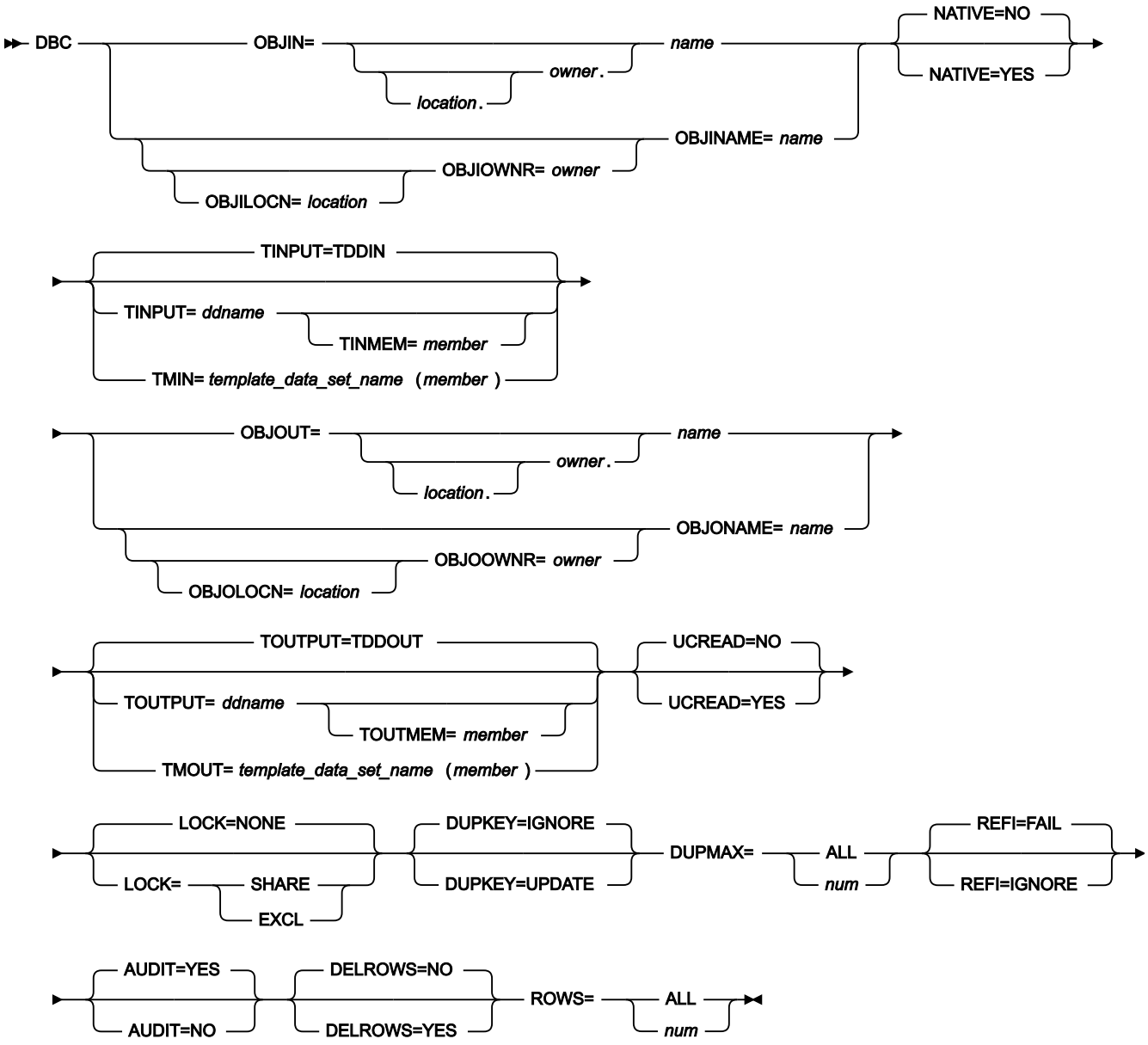
To copy data from one Db2® object to another Db2® object. Both objects need to be accessible from the currently connected Db2® system.

Usage

The ZDT/Db2 copy utility can:

- Optionally delete all rows from the target table prior to copying any data.
- Select the rows and columns to be copied.
- Limit the total number of rows to be copied.
- Reformat data into one or more new columns.
- Initialize columns with a new value or pattern.

The ZDT/Db2 copy utility takes no account of any referential integrity relationships defined on the source Db2® object.



OBJIN=*location.owner.name*

The optional name of the Db2® remote server (*location*) where the source object is located; the optional name of the owner of the source object (*owner*) and the source object name (*name*) to be copied.

When *location* is not specified the current (local) Db2® server is used. When the *owner* is not specified the object name is qualified using the current SQLID. When ZDT/Db2 generates the utility control statements, the *owner* value is non-blank.

OBJIN should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJILOCN, OBJIOWNR, OBJINAME keywords to specify the object.

OBJILOCN=*location*

The optional name of the Db2® remote server (*location*) where the source object is located. See Specifying a Db2 object name on page 1063.

OBJIOWNR=*owner*

The optional name of the owner of the source object (*owner*) to be copied. See Specifying a Db2 object name on page 1063.

OBJINAME=*name*

The object name (*name*) for the source object to be copied. See Specifying a Db2 object name on page 1063.

NATIVE

Specifies whether ZDT/Db2 should process Unicode encoded data natively when copying data between two Unicode encoded tables. This option can be ignored if the default processing of Unicode data is desired; that is, Unicode data is potentially substituted with error characters when copying data between Unicode tables.

YES

Process Unicode data natively.

NO

Do not Process Unicode data natively. This is the default.

TINPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the Db2® template that describes the source Db2® object to be copied. If you specify a concatenated DD, then you must provide the member name, *member*, via the TINMEM keyword. See Specifying the template for a Db2 object on page 1064.

TINMEM=*member*

The name of the template member in the dataset(s) identified by the TINPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TMIN parameter is specified. See Specifying the template for a Db2 object on page 1064.

TMIN=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the Z Data Tools Db2® template that describes the source Db2® object to be copied. See Specifying the template for a Db2 object on page 1064.

OBJOUT=*location.owner.name*

The optional name of the Db2® remote server (*location*) where the target object is located; the optional name of the owner of the target object (*owner*) and the target object name (*name*) to be copied.

When location is not specified the current (local) Db2® server is used. When the owner is not specified the object name is qualified using the current SQLID. When ZDT/Db2 generates the utility control statements, the owner value is non-blank.

OBJOUT should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJOLOCN, OBJOOWNR, OBJONAME keywords to specify the object.

OBJOLOCN=*location*

The optional name of the Db2® remote server (*location*) where the target object is located. See Specifying a Db2 object name on page 1063.

OBJOOWNR=*owner*

The optional name of the owner of the target object (*owner*) to be copied. See Specifying a Db2 object name on page 1063.

OBJONAME=*name*

The object name (*name*) for the target object to be copied. See Specifying a Db2 object name on page 1063.

TOUTPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the Db2® template that describes the target Db2® object. If you specify a concatenated DD, then you must provide the member name, *member*, via the TOUTMEM keyword. See Specifying the template for a Db2 object on page 1064.

TOUTMEM=*member*

The name of the template member in the dataset(s) identified by the TOUTPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TMOUT parameter is specified. See Specifying the template for a Db2 object on page 1064.

TMOUT=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the ZDT/Db2 template that describes the target Db2® object. See Specifying the template for a Db2 object on page 1064.

UCREAD

Specifies whether ZDT/Db2 should use uncommitted read, when accessing the source object:

YES

Uncommitted read is used. In this case, Db2® takes no locks when accessing the data in the source table, and it is possible that the data for a row changes after being read by ZDT/Db2, resulting in data inconsistencies.

NO

Uncommitted read is not used. This is the default.

LOCK

Specifies whether the source table should be locked, prior to copying data. The possible options are:

NONE

The default and recommended value. Any locks taken by Db2® depend on Db2® installation options, and options specified when the source object was created, unless negated by UCREAD=YES.

SHARE

ZDT/Db2 locks the source object in SHARE mode before copying data. This option is only effective for tables, it has no effect if the source object is a view. When an object is locked in SHARE mode, other users may read the object, but no updates by other users are possible.

EXCL

ZDT/Db2 locks the source object in EXCLUSIVE mode before copying the data. This option is only effective for tables, it has no effect if the source object is a view. When an object is locked in EXCLUSIVE mode, no other users may access the object at all, unless using uncommitted read.

DUPKEY

Specifies how ZDT/Db2 should respond to any duplicate key errors:

IGNORE

ZDT/Db2 ignores the error and continues processing. The row that caused the duplicate key error is not copied to the target Db2® object.

UPDATE

ZDT/Db2 attempts to update the existing row. The columns of the table that are part of the unique index that caused the SQLCODE-803 error are used to identify the row. No comparison is made between the copied row and the existing row prior to the update operation (that is, the update is done regardless of whether the copied row and existing row are the same). If there is more than one unique index defined on the target table, it is possible for an SQLCODE-803 to occur when the update is attempted. This is considered to be an update error.

DUPMAX

The number of duplicate key errors that are allowed before ZDT/Db2 terminates import processing:

ALL or 0

There is no limit.

num

The number of duplicate key errors allowed, up to a maximum of 2147483647.

REFI

Specifies whether to ignore errors arising from the violation of a check constraint (SQLCODE-545). The default is REFI=FAIL, meaning any check constraint error terminates copy processing. Otherwise (REFI=IGNORE) check constraint errors are ignored.

AUDIT

Specifies whether ZDT/Db2 is to write records to the ZDT/Db2 audit log:

YES

Write records to the ZDT/Db2 audit log. This is the default.

NO

Do not write records to the ZDT/Db2 audit log.

This option may be ignored if an installation audit option has been set that prevents a user modifying the installation setting.

DELROWS

Specifies whether ZDT/Db2 should attempt to delete all rows from the target table, prior to starting the copy operation:

YES

Attempt to delete all rows.

NO

Do not delete any rows. This is the default.

The deletion of rows from the target table is done using DELETE * FROM <object name> and fails if there are any referential integrity related errors arising from the deletion of rows.

ROWS

The maximum number of rows to be copied:

ALL

All rows from the source object are written to the target object.

num

The maximum number of rows to be copied. Valid range: 1-99999999.

Example

Examples

Example 1: Copy the DSN8810.EMP table to a remote location. Delete all rows in the target table before copying the data. Use uncommitted read when accessing DSN8810.EMP. Audit the access to DSN8810.EMP.

```
//DBC JOB (acct),'name'
//* Copy DSN8810.EMP table to the same table at a remote location
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
```

```
//SYSIN    DD *
$$FILEM DBC OBJIN="DSN8810"."EMP",
$$FILEM OBJOUT=REMLOC."DSN8810"."EMP",
$$FILEM     UCREAD=YES,
$$FILEM     LOCK=NONE,
$$FILEM     DUPKEY=IGNORE,
$$FILEM     DUPMAX=ALL,
$$FILEM     REFI=FAIL,
$$FILEM     AUDIT=YES,
$$FILEM     DELROWS=YES,
$$FILEM     ROWS=ALL
/*
```

Example 2: Copy the DSN8810.EMP table to a DSN8COPY.EMP. Update any rows in the target table that generate duplicate key errors, to a maximum of 100 duplicate key errors.

```
//DBC JOB (acct),'name'
//* Copy DSN8810.EMP table to a copy of the same table.
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM DBC OBJIN="DSN8810"."EMP",
$$FILEM OBJOUT="DSN8COPY"."EMP",
$$FILEM     LOCK=NONE,
$$FILEM     DUPKEY=UPDATE,
$$FILEM     DUPMAX=100,
$$FILEM     REFI=FAIL,
$$FILEM     ROWS=ALL
/*
```

Return codes

The return codes from the DBC function:

0

The function completed successfully

16

A serious error was encountered

Related functions

DBI

Import data from a sequential or VSAM file into a Db2® table (DBI (Import) batch command on page 1034)

DBX

Export data from a Db2® object to a sequential or VSAM file (DBX (Export) batch command on page 1043)

D2G

Create (generate) Db2® data (D2G (Data Generate) batch command on page 1058)

DBI (Import) batch command

Purpose

To import data, in a variety of formats, from a QSAM or VSAM file, into a Db2® object. The Db2® object must be accessible from the currently connected Db2® system.

Usage

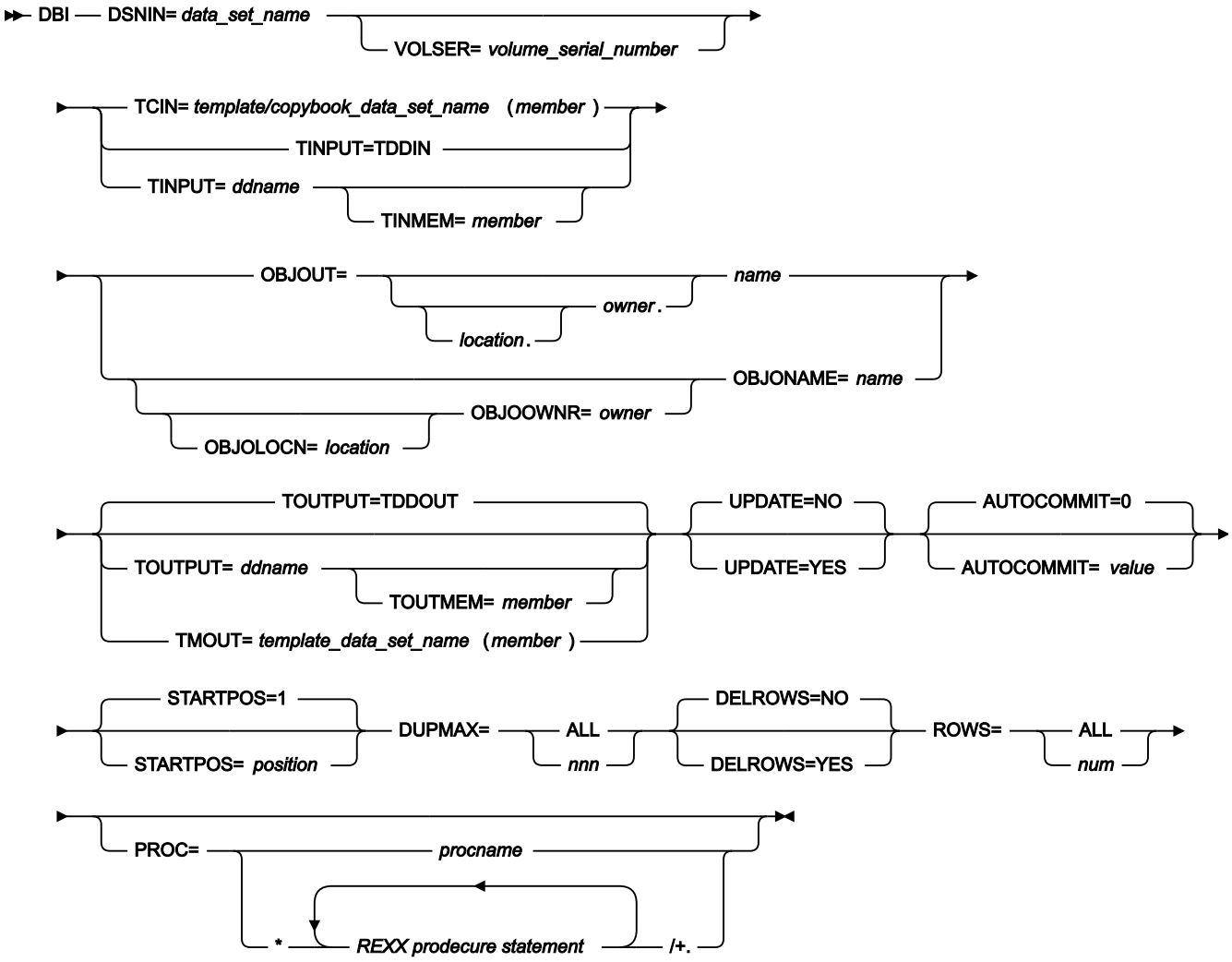
The ZDT/Db2 import utility can:

- Optionally delete all rows from the target table prior to importing any data.
- Limit the number of records to be imported.
- Select the records to be imported.
- Map fields (using template mapping) in the import data set to columns in the target table, with appropriate data type and length conversions.
- Optionally update existing rows, when the values in an imported row's unique index columns match an existing table row.

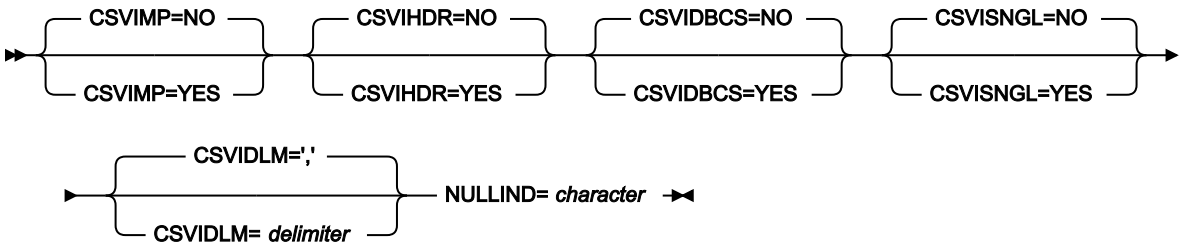
A REXX procedure can be used during the import to modify mapped and converted data, prior to inserting it into the target Db2® table. Any REXX functions can be used at this point, and records can be dropped, or the import stopped via the REXX proc.

At the end of the import process the following statistics are shown:

- The count of rows successfully imported.
- The number of duplicate row errors.
- Optionally, the number of duplicate rows updated.
- Optionally, the number of duplicate row updates that failed.
- The number of rows in error.
- The number of rows skipped via template selection.
- The number of rows dropped by the REXX proc.



CSV import options



DSNIN=*data_set_name*

The data set name containing the data to be imported. If the data set is partitioned, a member name is required.

VOLSER=*volume_serial_number*

The volume serial number for the input data set. This is only required for uncataloged data sets.

TINPUT=*ddname*

Defines a reference to a DD statement for the data sets which contains the template member that describes the record structure of your input data. The default is TDDIN.

TINMEM=*member*

The name of the template member in the dataset(s) identified by the TINPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TCIN parameter is specified.

TCIN=*template/copybook_data_set_name*(*member*)

The PDS (*template/copybook_data_set_name*) and member name (*member*) of the template or copybook that describes the record structure of the data to be imported. The JCL generated by ZDT/Db2 specifies TCIN. See Specifying the template/copybook for a data set on page 1064.

OBJOUT=*location.owner.name*

The optional name of the Db2® remote server (*location*) where the target object is located; the optional name of the owner of the target object (*owner*) and the target object name (*name*) to be copied.

When location is not specified the current (local) Db2® server is used. When the owner is not specified the object name is qualified using the current SQLID. When ZDT/Db2 generates the utility control statements, the owner value is non-blank.

OBJOUT should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJOLOCN, OBJOOWNR, OBJONAME keywords to specify the object.

OBJOLOCN=*location*

The optional name of the Db2® remote server (*location*) where the target object is located. See Specifying a Db2 object name on page 1063.

OBJOOWNR=*owner*

The optional name of the owner of the target object (*owner*) for the import. See Specifying a Db2 object name on page 1063.

OBJONAME=*name*

The object name (*name*) for the target object for the import. See Specifying a Db2 object name on page 1063.

TOUTPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the Db2® template that describes the target Db2® object. If you specify a concatenated DD, then you must provide the member name, *member*, via the TOUTMEM keyword. See Specifying the template for a Db2 object on page 1064.

TOUTMEM=*member*

The name of the template member in the dataset(s) identified by the TOUTPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TMOUT parameter is specified. See Specifying the template for a Db2 object on page 1064.

TMOUT=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the ZDT/Db2 template that describes the target Db2® object. See Specifying the template for a Db2 object on page 1064.

UPDATE

Specifies whether ZDT/Db2 should attempt to update any rows that generate SQLCODE-803 (duplicate key) errors:

YES

ZDT/Db2 attempts to update the existing row. All columns of the table that are part of at least 1 unique index are used to identify the row. No comparison is made between the imported row and the existing row prior to the update operation (that is, the update is done regardless of whether the imported row and existing row are the same). When using this option it is recommended that a Db2® table name, rather than a view name, be specified. ZDT/Db2 does not attempt to update existing rows in a view defined on a view.

NO

No rows are updated. This is the default.

AUTOCOMMIT=*value*

Specifies a numeric value indicating how often ZDT/Db2 should issue a Db2® commit during the import. The default is zero, meaning that no Db2® commits are issued during the import operation. Any positive value results in ZDT/Db2 issuing a Db2® commit after value successful changes are made during the import. A successful change occurs when either a record is successfully inserted, or when a row is successfully updated after the attempted insert of the record resulted in a duplicate key error. The counter is reset after each Db2® commit is issued, therefore multiple Db2® commits may be issued. If an error occurs during the import and one or more Db2® commits have been issued, only the changes made subsequent to the last Db2® commit point are backed out.

STARTPOS=*position*

Specifies a numeric value indicating the starting record for the import. The default is 1, meaning ZDT/Db2 starts the import with the first record of the import file. Any value greater than 1 causes ZDT/Db2 to skip *position*-1 records before commencing the import.

DUPMAX

The number of duplicate key errors that are allowed before ZDT/Db2 terminates import processing:

ALL or 0

There is no limit.

num

The number of duplicate key errors allowed, up to a maximum of 2147483647.

DELROWS

Specifies whether ZDT/Db2 should attempt to delete all rows from the target table, prior to starting the copy operation:

YES

Attempt to delete all rows.

NO

Do not delete any rows. This is the default.

The deletion of rows from the target table is done using DELETE * FROM <object name> and fails if there are any referential integrity related errors arising from the deletion of rows.

ROWS

The maximum number of records to be imported:

ALL

All records from the import data set are inserted into the target object.

num

The maximum number of records to be imported. Valid range: 1-99999999.

PROC

REXX procedure:

procname

The name of the REXX procedure to be used when processing import records.

*****

An inline procedure. The * is followed by one or more REXX procedure statements, and the terminating "/+.".

CSV Import Options

CSVIMP

Default NO. Specify CSVIMP=YES when importing CSV data with a template.

CSVIHDR

Default NO. Specify YES if the CSV data set to be imported includes headings.

CSVIDBCS

Default NO. Specify YES if the CSV data set to be imported includes double-byte data (DBCS) characters wrapped in shift-in (x'0E') and shift-out (x'0F') characters.

CSVISNGL

Default NO. Specify YES option if the CSV data set to be imported has single quote encapsulation.

CSVIDLM=*delimiter*

Default is the comma "," character. Use this keyword to specify an alternate delimiter value to the comma "," character.

NULLIND=*character*

Use this keyword to specify the null indicator value that was used to represent a null value in the CSV input data set. A nullable column will be set to null when the corresponding CSV value has:

1. A non-blank null indicator value.
2. A null CSV value. For example, two successive delimiters ',,'.

Example

Examples

Example 1: Import data from a sequential file into DSN8810.EMP. Delete all rows in DSN8810.EMP before importing the data

```
//DBI JOB (acct),'name'
//* Import data from a sequential file into DSN8810.EMP.
//* Delete all the rows from the table before importing the data.
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM DBI DSNIN=ID1.EXPORT.EMP.EXAMPLE1,
$$FILEM TCIN=ID1.HFM.TEMPLATE(EMP),
$$FILEM OBJOUT="DSN8810"."EMP",
$$FILEM UPDATE=NO,
$$FILEM DUPMAX=ALL,
$$FILEM DELROWS=YES,
$$FILEM ROWS=ALL
/*
```

Example 2: Import data from a sequential file into DSN8810.EMP. Use REXX proc PROC1 to process each record in the sequential file before inserting the record. Update any rows that cause duplicate key errors, to a maximum of 50 duplicate key errors.

```
//DBI JOB (acct),'name'
//* Import data from a sequential file into DSN8810.EMP.
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//HFMEXEC  DD DSN=ID1.EXEC,DISP=SHR
```

```
//SYSIN    DD *
$$FILEM DBI DSNIN=ID1.EXPORT.EMP.EXAMPLE1,
$$FILEM TCIN=ID1.HFM.TEMPLATE(EMP),
$$FILEM OBJOUT="DSN8810"."EMP",
$$FILEM UPDATE=YES,
$$FILEM DUPMAX=50,
$$FILEM DELROWS=NO,
$$FILEM ROWS=ALL,
$$FILEM PROC=PROC1
/*
```

Return codes

The return codes from the DBI function:

0

The function completed successfully

16

A serious error was encountered

Related functions

DBC

Copy data from one Db2® object to another (DBC (Copy) batch command on page 1027)

DBX

Export data from a Db2® object to a sequential or VSAM file (DBX (Export) batch command on page 1043)

D2G

Create (generate) Db2® data (D2G (Data Generate) batch command on page 1058)

DBP (Print) batch command

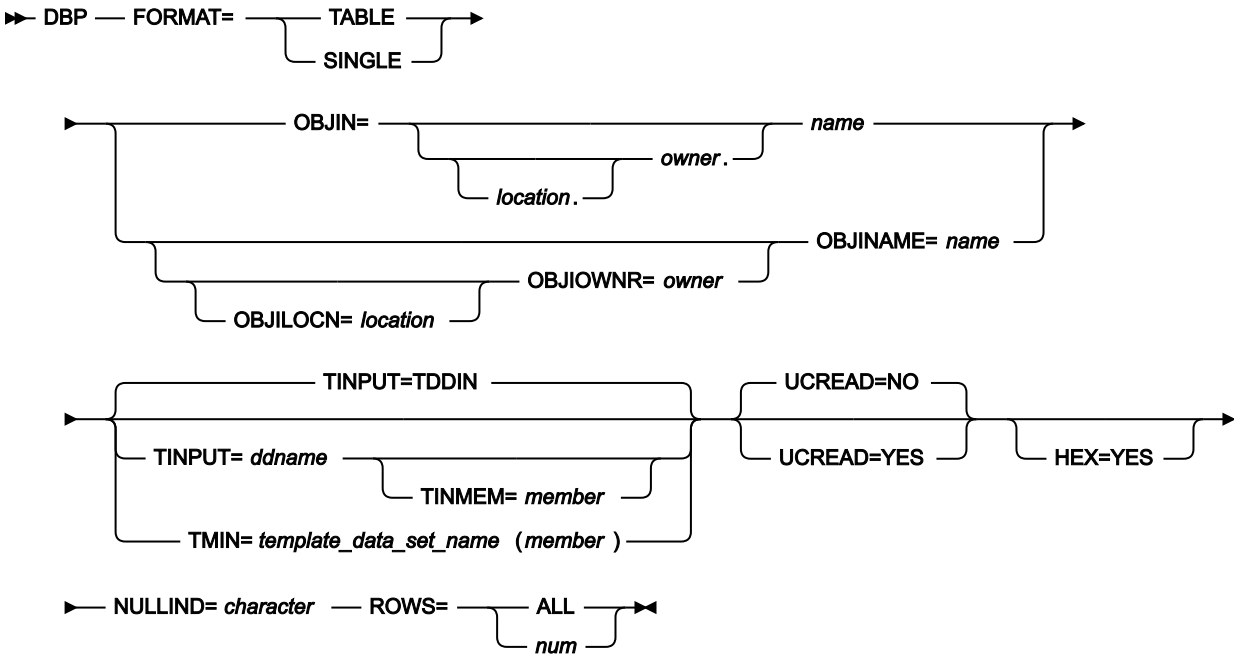
Purpose

To print the contents of a Db2® object in a variety of formats.

Usage

The ZDT/Db2 Print utility function can:

- Print data in table or single formats, in either character or HEX.
- Select the rows and columns to be printed.
- Limit the number of rows to be printed.



FORMAT

The formatting mode for the printed data:

TABLE

In table mode each row of the table occupies a single line, and columns appear left to right across the page.

SINGLE

In single mode each row of the table occupies multiple print lines. The first column of the object is printed on the first, and possibly more lines, followed by the second column and so on.

OBJIN=*location.owner.name*

The optional name of the Db2® remote server (*location*) where the object is located; the optional name of the owner of the object (*owner*) and the object name (*name*) to be printed.

When location is not specified the current (local) Db2® server is used. When the owner is not specified the object name is qualified using the current SQLID. When ZDT/Db2 generates the utility control statements, the owner value is non-blank.

OBJIN should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJILOCN, OBJIOWNR, OBJINAME keywords to specify the object.

OBJILOCN=*location*

The optional name of the Db2® remote server (*location*) where the object to be printed is located. See Specifying a Db2 object name on page 1063.

OBJIOWNR=*owner*

The optional name of the owner of the object (*owner*) to be printed. See Specifying a Db2 object name on page 1063.

OBJINAME=*name*

The object name (*name*) for the object to be printed. See Specifying a Db2 object name on page 1063.

TINPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the Db2® template that describes the source Db2® object to be printed. If you specify a concatenated DD, then you must provide the member name, *member*, via the TINMEM keyword. See Specifying the template for a Db2 object on page 1064.

TINMEM=*member*

The name of the template member in the dataset(s) identified by the TINPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TMIN parameter is specified. See Specifying the template for a Db2 object on page 1064.

TMIN=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the Z Data Tools template that describes the source Db2® object to be copied. See Specifying the template for a Db2 object on page 1064.

UCREAD

Specifies whether ZDT/Db2 should use uncommitted read, when accessing the source object:

YES

Uncommitted read is used. In this case, Db2® takes no locks when accessing the data in the source table, and it is possible that the data for a row changes after being read by ZDT/Db2, resulting in data inconsistencies.

NO

Uncommitted read is not used. This is the default.

HEX=YES

Specifies that ZDT/Db2 should print the data in hexadecimal format. Omit this option if normal printing is required.

NULLIND=*character*

Defines the character used to indicate that the value for a column is the Db2® null value, in the printed data. The default is the underscore character. The value for this character is set using the editor **Null Column Indicators: Display** option. When specified in the Z Data Tools utility control statements, the character must be a printable character other than a comma. The value cannot be enclosed in quotation marks.

ROWS

The maximum number of rows to be printed:

ALL

All rows from the source object are printed.

num

The maximum number of rows to be printed. Valid range: 1-99999999.

Example

Examples

Example 1: Print data from DSN8810.EMP, in single mode. Use uncommitted read when accessing the Db2® data. Print the data in hexadecimal format.

```
//DBP JOB (acct),'name'
//* Print table DSN8810.EMP in SINGLE format.
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM DBP FORMAT=SINGLE,
$$FILEM OBJIN="DSN8810"."DEPT",
$$FILEM UCREAD=YES,
$$FILEM HEX=YES,
$$FILEM NULLIND=_,
$$FILEM ROWS=ALL
/*
```

Return codes

The return codes from the DBP function:

0

The function completed successfully

16

A serious error was encountered

Related functions

None.

DBX (Export) batch command

Purpose

To export data from a Db2® object (table, view, synonym or alias) or the result table from a SELECT statement, in any local or distributed Db2® system, to a sequential or VSAM file.

Usage

The ZDT/Db2 Export Utility can export data from a Db2® table or view using one of the following methods:

Z Data Tools

The output data set can be a sequential data set, a partitioned data set, or a VSAM file.

The Db2® UNLOAD utility

The output data set is constrained by the Db2® UNLOAD utility to be a sequential data set. Any reformatting of the output data is performed by the Db2® UNLOAD utility. Therefore some template mapping and re-formatting options that are available when using Z Data Tools are not available.

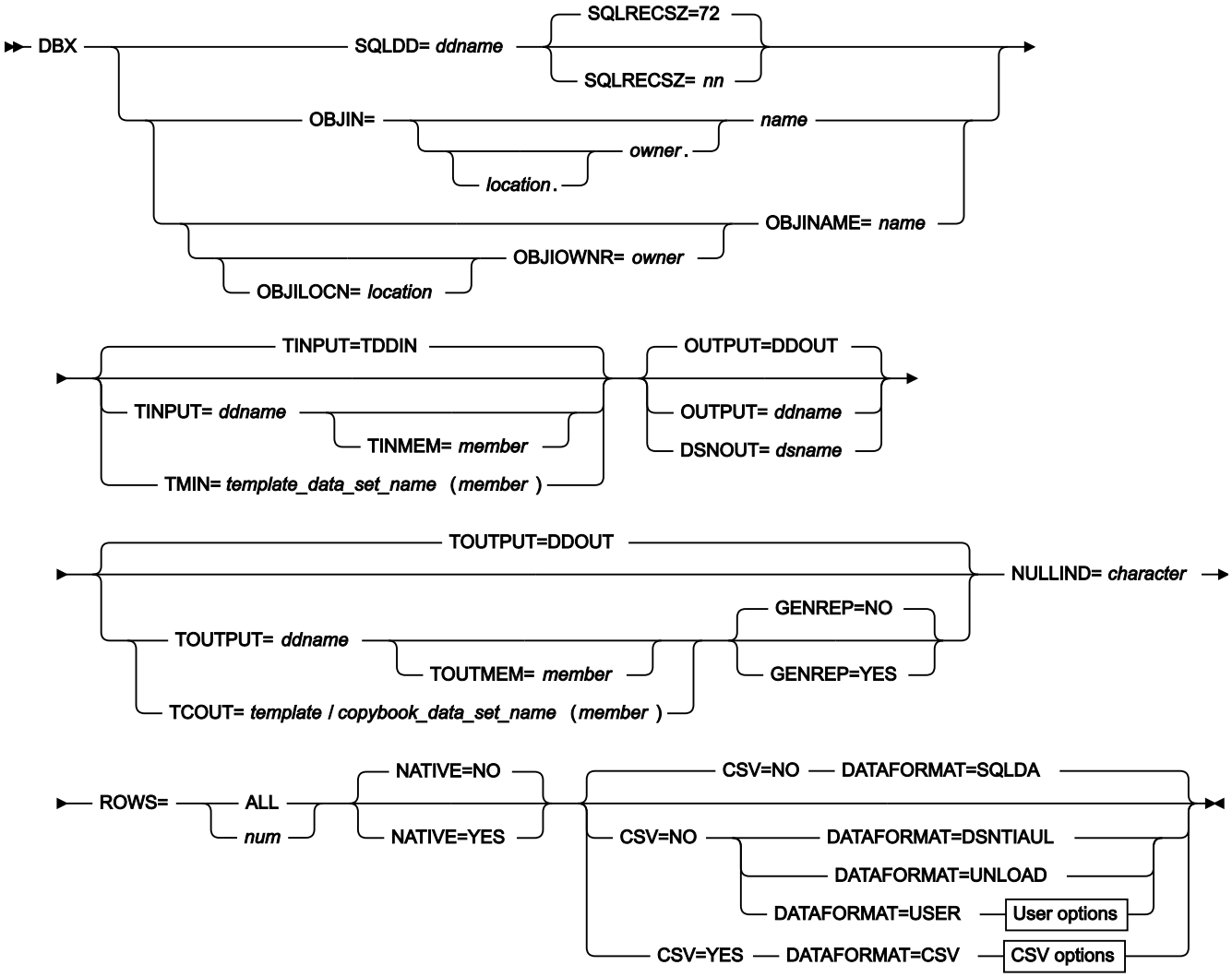
The Db2® DSNTIAUL sample program

The output data set is constrained by the Db2® DSNTIAUL utility to be a sequential data set. Any reformatting of the output data is performed by the Db2® DSNTIAUL sample program utility. Therefore some template mapping and re- formatting options that are available when using Z Data Tools are not available.

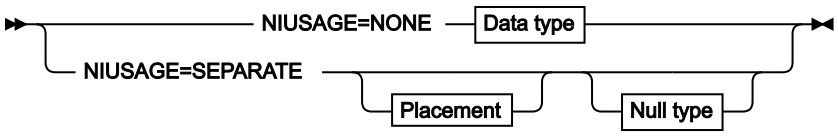
When exporting large volumes of data, the Db2® UNLOAD utility offers superior performance when compared with Z Data Tools or the DSNTIAUL sample program.

You can choose a pre-defined (standard) export data format, or specify the data format using a copybook or template. You can specify that Z Data Tools is to export the data, or use a Db2® utility or sample program to export the data. For a user-defined data format you can specify how the Db2® null indicator is represented (either one byte or two byte) and where the null indicator is located in the exported data. You can also specify that the data is to be exported in delimited format (comma separated variables).

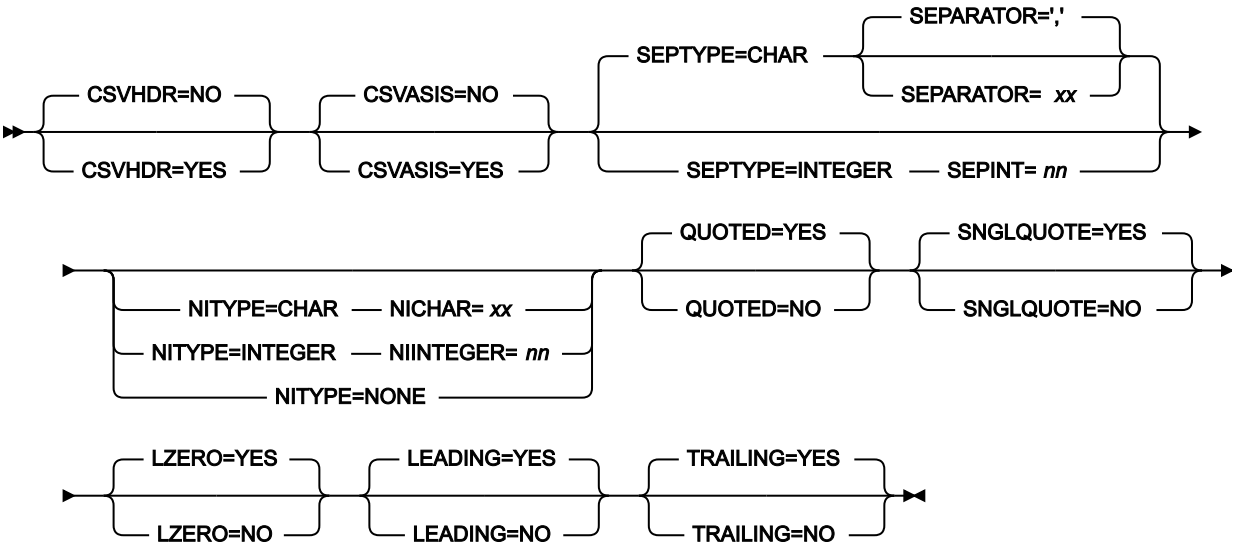
Figure 181.



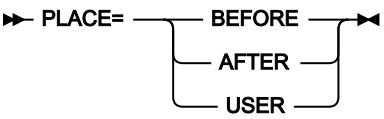
User options



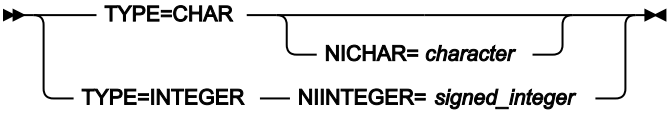
CSV options



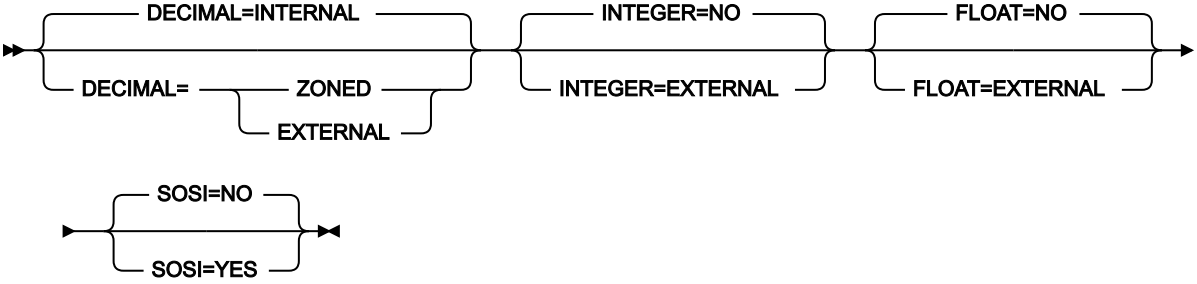
Placement



Null type



Data type



SQLDD=*ddname*

ddname is the DDNAME, specified in the JCL, that identifies either a data set containing an SQL SELECT statement to be processed, or a DDNAME used to specify an in-line SQL SELECT statement.

SQLRECSZ=*nn*

nn is the usable record size for the SQL statement. For in-line SQL statements, specified in the JCL, use a value of 72, to avoid problems with sequence numbers in columns 73-80 inclusive.

For a data set, set an appropriate value.

OBJIN=*location.owner.name*

The optional name of the Db2® remote server (*location*) where the object is located; the optional name of the owner of the object (*owner*) and the object name (*name*) to be exported.

When location is not specified the current (local) DB2® server is used. When the owner is not specified the object name is qualified using the current SQLID. When FM/DB2 generates the utility control statements, the owner value is non-blank.

OBJIN should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJILOCN, OBJIOWNR, OBJINAME keywords to specify the object.

OBJILOCN=*location*

The optional name of the Db2® remote server (*location*) where the object to be exported is located. See Specifying a Db2 object name on page 1063.

OBJIOWNR=*owner*

The optional name of the owner of the object (*owner*) to be exported. See Specifying a Db2 object name on page 1063.

OBJINAME=*name*

The object name (*name*) for the object to be exported. See Specifying a Db2 object name on page 1063.

TINPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the Db2® template that describes the Db2® object to be exported. If you specify a concatenated DD, then you must provide the member name, *member*, via the TINMEM keyword. See Specifying the template for a Db2 object on page 1064.

TINMEM=*member*

The name of the template member in the datasets identified by the TINPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TMIN parameter is specified. See Specifying the template for a Db2 object on page 1064.

TMIN=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the ZDT/Db2 template that describes the Db2® object to be exported. See Specifying the template for a Db2 object on page 1064.

OUTPUT=*ddname*

Defines a reference to a DD statement for the export data set. This keyword and the DSNOUT keyword are mutually exclusive. The JCL generated by ZDT/Db2 specifies OUTPUT=FMNOUT in the Z Data Tools utility control statements, and an appropriate DD statement in the JCL.

DSNOUT=*dsname*

Defines the name of the export data set or an absolute path to the export z/FS file (directory). This keyword and the OUTPUT keyword are mutually exclusive. The name may include a member name in parenthesis. An absolute path to an z/FS file (directory) must be enclosed in apostrophes. If the data set name does not fit on one line, you can split it over more than one line.

TOUTPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the template member that describes the record structure of your output data. The default is TDDOUT.

TOUTMEM=*member*

The name of the template member in the dataset(s) identified by the TOUTPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TCOUT parameter is specified.

TCOUT=*template/copybook_data_set_name*(*member*)

The PDS (*template/copybook_data_set_name*) and member name (*member*) of the template or copybook that describes the record structure of the exported data. The JCL generated by ZDT/Db2 specifies TCOUT, when an output copybook or template is required. See Specifying the template/copybook for a data set on page 1064.

GENREP

Whether to generate and save the output template to the data set specified on the TCOUT keyword or TDDOUT DD statement.

NO

Don't generate and save.

YES

Generate and save.

NULLIND=*character*

For CSV format only, defines the character used to indicate that the value for a column is the Db2® null value, in the exported data. The default is the underscore character. The value for this character is set using the editor **Null Column Indicators: Display** option. When specified in the Z Data Tools utility control statements, the character must be a printable character other than a comma. The value cannot be enclosed in quotation marks.

NULLIND is always specified in the Z Data Tools utility control statements. The value specified is ignored for all DATAFORMAT specifications other than CSV.

ROWS

The number of records to be exported:

ALL

All records selected from the object are written to the export data set.

num

The maximum number of records to be written to the export data set. Valid range: 1-99999999.

NATIVE

Specifies whether ZDT/Db2 should process Unicode encoded data natively when exporting data from a Unicode encoded table. This option can be ignored if the default processing of Unicode data is desired; that is, Unicode data is potentially substituted with error characters when copying data between Unicode tables.

YES

Process Unicode data natively.

NO

Do not Process Unicode data natively. This is the default.

CSV

Specifies whether the exported data is to be in delimited variables format or not:

NO

Export data is not to be in delimited format. This is the default.

YES

Export data is to be in delimited format.

DATAFORMAT

The data format for the exported data:

SQLDA

The default. Any nullable column is preceded by a 2-byte binary field containing the null indicator. This field contains -1 if the column has the null value and 0 otherwise. Any decimal field occupies 16 bytes, with the data left-justified in the field and trailing binary zeros. Any graphic or vargraphic field is followed by two bytes containing binary zeros.

UNLOAD

In this format any nullable column is preceded by a 1-byte field containing the null indicator. This field contains hexadecimal "FF" if the column had a null value and 0 otherwise. Any decimal field occupies the minimum number of bytes needed to represent the data as a packed decimal field. Variable length columns in the unloaded records are padded to their maximum length and the padded data fields are preceded by two-byte length fields that indicate the size of the actual data without the padding. CCSID conversion is not supported. The exported data is in the CCSID used to bind the ZDT/Db2 plan.

DSNTIAUL

Any nullable column is followed by a 1-byte field containing the null indicator. This field contains "?" if the column had a null value. Any decimal field occupies the minimum number of bytes needed to represent the data as a packed decimal field.

USER

The data format is defined by a user-specified copybook or template and template mapping. Alternatively the data format can be defined using the NIUSAGE, PLACE, TYPE (NICHAR, NIINTEGER), DECIMAL, INTEGER, FLOAT, and SOSI keywords.

CSV

The data is exported as delimited variables. The default delimiter is the comma, the SEPARATOR keyword can be used to specify a different delimiter. A copybook/template cannot be used to describe the delimited variables data format.

CSVHDR

Specifies whether column headers are to be produced on the first record of the export file when CSV=YES has been specified. Possible values are:

NO

Column headers not produced.

YES

Column headers produced.

CSVASIS

Specifies whether the automatic encapsulation for CSV data is disabled when the data contains the CSV separator character or a quote.

YES

Automatic encapsulation is disabled.

NO

Automatic encapsulation is enabled.

This option applies to character data other than pure DBCS data (GRAPHIC datatypes). One use of this option is to prevent double encapsulation of character data that has already been encapsulated with quotes before insertion into a DB2® table.

SEPTYPE=CHAR/INTEGER

Specifies, for CSV format only, the type of separator used to separate each field in the output.

CHAR

One or two characters is used to separate data in each field. Use the SEPARATOR keyword to specify the characters to be used.

INTEGER

A signed integer is used to specify a 2-byte separator. Use SEPINT to specify the signed integer to be used.

SEPARATOR='*characters*'

Specifies the one or two characters used to separate each field when exporting to CSV format. The default is the comma.

SEPINT=*signed_integer*

Specifies a signed integer used to separate each field when exporting to CSV format. The value specified is converted to the 2-byte signed representation of the integer. Examples:

`-1` results in a two-byte separator X'FFFF'
`257` results in a two-byte separator X'0101'

QUOTED

Specifies whether output field values are to be enclosed in quotes. The default is QUOTED=NO. If this option is specified, and SNGLQUOTE is not specified, or, SNGLQUOTE is set to NO, then output fields will be enclosed in double quotes.

YES

Enclose output field values in quotes.

NO

Default. Do not enclose output field values in quotes.

SNGLQUOTE

Specifies whether output fields values are to be enclosed in single quotes or double quotes. The default is SNGLQUOTE=NO.

YES

Enclose output field values in single quotes.

NO

Default. Enclose output field values in double quotes.

LZERO

Specifies whether numeric fields will be prefixed with leading zeros. The default is LZERO=NO.

YES

Prefix numeric fields with leading zeros.

NO

Default. Do not prefix numeric fields with leading zeros.

LEADING

Specifies if leading blanks within Db2 CHAR and VARCHAR columns are to be included in the CSV export data set. The default is LEADING=YES.

YES

Include leading blanks in the exported data set. If the value of the Db2 column is all blanks, and TRAILING=YES is specified, then all of the blanks are included in the exported data set.

NO

Do not include leading blanks in the exported data set.

TRAILING

Specifies if trailing blanks within Db2 CHAR and VARCHAR columns are to be included in the CSV exported data set. The default is TRAILING=NO.

YES

Include trailing blanks in the exported data set. If the value of the Db2 column is all blanks, and LEADING=YES is specified, then all of the blanks are included in the exported data set.

NO

Do not include trailing blanks in the exported data set.

PLACE

The location of any Db2® null indicator fields in the exported data:

BEFORE

The field for the Db2® null indicator appears immediately before the data for the column.

AFTER

The field for the Db2® null indicator appears immediately after the data for the column.

USER

The field for the Db2® null indicator is defined using a copybook or template, with template mapping.

NIUSAGE

Specifies, for DATAFORMAT=USER only, whether or not null indicators appear in the exported data. To specify null indicator usage for CSV format use the NITYPE, NICHAR, and NIINTEGER keywords.

NONE

The exported data is not to contain any DB2® null indicators. Using this option with a DB2® object that has nullable columns results in a loss of information, in that null indicators in the source record are not exported.

SEPARATE

The exported data is to contain DB2® null indicators. The position and data type of the null indicator fields is specified using the PLACE, TYPE, NICHAR, and NIINTEGER keywords.

TYPE=CHAR/INTEGER

The data format for the null indicator fields, for DATAFORMAT=USER only. The null indicator can be either a one-byte (character) or two-byte (signed integer) field. The character or integer used to indicate the DB2® null value can be specified using the NICHAR (TYPE=CHAR) or NIINTEGER (TYPE=INTEGER) keywords.

NITYPE=CHAR/INTEGER

The data format for the null indicator fields, for DATAFORMAT=CSV only. The null indicator can be either a one-byte (character) or two-byte (signed integer) field. The character or integer used to indicate the DB2® null value can be specified using the NICHAR (NITYPE=CHAR) or NIINTEGER (NITYPE=INTEGER) keywords.

NICHAR=*character*

The character used to indicate that a column has the DB2® null value. NICHAR should only be specified with DATAFORMAT=USER and TYPE=CHAR.

NIINTEGER=*signed_integer*

The integer used to indicate that a column has the DB2® null value. NIINTEGER should only be specified with DATAFORMAT=USER and TYPE=INTEGER. You can specify any signed integer in the range -32767 to 32768.

DECIMAL

Specifies the data format for any decimal columns in the exported data. Possible values are:

INTERNAL

Decimal data appears in internal packed decimal format.

ZONED

Decimal data appears as a string of zoned decimal digits.

EXTERNAL

Decimal data appears as a string of characters, the string represents the decimal number, for example '3.1415'.

INTEGER

Specifies the data format for any integer columns in the exported data. An integer column can be a SMALLINT, INTEGER or BIGINT. Possible values are:

NO

Integer data appears in internal format, occupying 2 bytes (SMALLINT), 4 bytes (INTEGER) or 8 bytes (BIGINT).

EXTERNAL

Integer data appears as a string of characters, the string represents the integer, for example, '42', '1000000'.

FLOAT

Specifies the data format for any floating point columns in the exported data. A floating point column is a REAL, FLOAT, DOUBLE or DECFLOAT column. Possible values are:

NO

The floating point data appears in internal format.

EXTERNAL

Floating point data appears as a string of characters, the string represents the number, for example '-1.0000000000000000E+00'.

SOSI

Specifies whether the data for graphic data type columns (GRAPHIC, VARGRAPHIC) should be enclosed in shift-out/shift-in (sosi) characters. Possible values are:

NO

The graphics data is not enclosed in sosi characters.

YES

The graphics data is enclosed in sosi characters.

Example

Examples

Example 1: Export data from Db2® sample table DSN8810.EMP to a sequential file, using SQLDA format. The output data set is allocated in the batch job.

```
//DBX JOB (acct),'name'
//* Export DSN8810.EMP table to a data set
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=(MOD,CATLG,DELETE),UNIT=SYSALLDA,
//            AVGREC=U,
//            LRECL=144,RECFM=FB,DSORG=PS,
//            SPACE=(144,(42,4),RLSE),
//            DSN=ID1.EXPORT.EMP.SQLDA
//SYSIN    DD *
$$FILEM DBX OBJIN="DSN8810"."EMP",
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=NO,
$$FILEM DATAFORMAT=FMSQLDA,
$$FILEM ROWS=ALL
/*
```

Example 2: Export data from Db2® sample table DSN8810.EMP to a sequential file, using Db2® UNLOAD format. The output data set is allocated in the batch job.

```
//DBX JOB (acct),'name'
//* Export DSN8810.EMP table to a data set
```

```
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=(MOD,CATLG,DELETE),UNIT=SYSALLDA,
//            AVGREC=U,
//            LRECL=101,RECFM=FB,DSORG=PS,
//            SPACE=(101,(42,4),RLSE),
//            DSN=ID1.EXPORT.EMP.UNLOAD
//SYSIN    DD *
$$FILEM DBX OBJIN="DSN8810"."EMP",
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=NO,
$$FILEM DATAFORMAT=UNLOAD,
$$FILEM ROWS=ALL
/*
```

Example 3: Export data from Db2® sample table DSN8810.EMP to a sequential file, using Db2® DSNTIAUL format. The output data set is allocated in the batch job.

```
//DBX JOB (acct),'name'
//* Export DSN8810.EMP table to a data set
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=(MOD,CATLG,DELETE),UNIT=SYSALLDA,
//            AVGREC=U,
//            LRECL=101,RECFM=FB,DSORG=PS,
//            SPACE=(101,(42,4),RLSE),
//            DSN=ID1.EXPORT.EMP.DSNTIAUL
//SYSIN    DD *
$$FILEM DBX OBJIN="DSN8810"."EMP",
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=NO,
$$FILEM DATAFORMAT=DSNTIAUL,
$$FILEM ROWS=ALL
/*
```

Example 4: Export data from a table with a long name containing spaces to a sequential file, using SQLDA format. The output data set is pre-allocated.

```
//DBX JOB (acct),'name'
//* Export a table with a long name to a data set
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V910.DSN1.SDSNEXIT,DISP=SHR
```

```
//         DD DSN=DB2.V910.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=MOD,
//            DSN=ID1.EXPORT.EMP.EXAMPLE4
//SYSIN    DD *
$$FILEM DBX OBJIOWNR=ID1,
$$FILEM OBJINAME='\$Very long table name to demonstrate how \$',
$$FILEM '\$long names are handled\$',
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=NO,
$$FILEM DATAFORMAT=FMSQLDA,
$$FILEM ROWS=ALL
/*
```

Example 5: Export data from Db2® sample table DSN8810.EMP to a sequential file, using a user-defined data format. The output data set is pre-allocated. In the user-defined data format:

- Null indicators are used (NIUSAGE=SEPARATE)
- Null indicators appear immediately after the data for the column (PLACE=AFTER)
- The null indicator is a two byte (signed integer) field (TYPE= INTEGER)
- The value used to indicate a column has the Db2® null value is -55.
- Any DECIMAL data type columns appear in external format (for example '3.1414')
- Any INTEGER data columns appear in external format (for example '1234')
- Any floating point data type columns appear in external format.

```
//DBX JOB (acct),'name'
//* Export DSN8810.EMP table to a data set
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V910.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V910.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=MOD,
//            DSN=ID1.EXPORT.EMP.EXAMPLE5
//SYSIN    DD *
$$FILEM DBX OBJIN="DSN8910"."EMP",
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=NO,
$$FILEM DATAFORMAT=USER,
$$FILEM NIUSAGE=SEPARATE,
$$FILEM PLACE=AFTER,
$$FILEM TYPE=INTEGER,
$$FILEM NIINTEGER=-55,
$$FILEM DECIMAL=EXTERNAL,
$$FILEM INTEGER=EXTERNAL,
$$FILEM FLOAT=EXTERNAL,
```

```
$$FILEM ROWS=ALL
/*
```

Example 6: Export data from Db2® sample table DSN8810.EMP to a sequential file, using a CSV (delimited variables) format. The delimiter character is the '!' character. The output data set is pre- allocated.

```
//DBX JOB (acct),'name'
//* Export DSN8810.EMP table to a data set
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V910.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V910.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=MOD,
//            DSN=ID1.EXPORT.EMP.CSV
//SYSIN    DD *
$$FILEM DBX OBJIN="DSN8910"."EMP",
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=YES,
$$FILEM DATAFORMAT=CSV,
$$FILEM SEPARATOR='!',
$$FILEM ROWS=ALL
/*
```

Example 7: Export data from Db2® sample table DSN8810.EMP to a sequential file, using a CSV (delimited variables format). The delimiter character is x'FFFF', defined using the NIUSAGE, TYPE and NIINTEGER keywords. The output data set is pre-allocated.

```
//DBX JOB (acct),'name'
//* Export DSN8810.EMP table to a data set
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V910.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V910.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//FMNOUT   DD DISP=MOD,
//            DSN=ID1.EXPORT.EMP.CSV2
//SYSIN    DD *
$$FILEM DBX OBJIN="DSN8910"."EMP",
$$FILEM OUTPUT=FMNOUT,
$$FILEM NULLIND=_,
$$FILEM CSV=YES,
$$FILEM DATAFORMAT=CSV,
$$FILEM NIUSAGE=CSV,
$$FILEM TYPE=INTEGER,
$$FILEM NIINTEGER=-1,
$$FILEM ROWS=ALL
/*
```

Return codes

The return codes from the DBX function:

0

The function completed successfully

16

A serious error was encountered

Related functions

DBI

Import data from a sequential or VSAM file into a Db2® table (DBI (Import) batch command on page 1034)

DBC

Copy data from one Db2® object to another (DBC (Copy) batch command on page 1027)

D2G

Create (generate) Db2® data (D2G (Data Generate) batch command on page 1058)

D2G (Data Generate) batch command

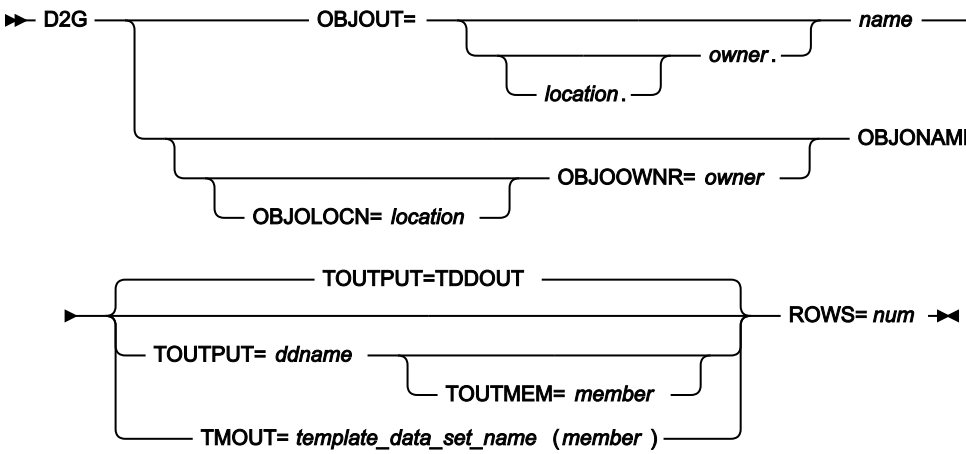
Purpose

To load a Db2® object with test data.

Usage

The ZDT/Db2 Data Create utility function can:

- Limit the number of rows to be created.
- Specify how, on a column-by-column basis, each row is be initialized.



OBJOUT location.owner.object

The optional name of the Db2® remote server (location) where the object is located; the optional name of the owner of the object (owner) and the object name (name).

When location is not specified the current (local) Db2® server is used. When the owner is not specified the object name is qualified using the current SQLID. When ZDT/Db2 generates the utility control statements, the owner value is non-blank.

OBJOUT should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJOLOCN, OBJOOWNR, OBJONAME keywords to specify the object.

OBJOLOCN=*location*

The optional name of the Db2® remote server (*location*) where the object is located. See Specifying a Db2 object name on page 1063.

OBJOOWNR=*owner*

The optional name of the owner of the object (*owner*). See Specifying a Db2 object name on page 1063.

OBJONAME=*name*

The object name (*name*) for the object. See Specifying a Db2 object name on page 1063.

TMOUT=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the ZDT/Db2 template that describes the Db2® object where the data is to be created. See Specifying the template for a Db2 object on page 1064. The JCL generated by ZDT/Db2 always specifies TMOUT.

TOUTPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the template member that describes the record structure of your output data. The default is TDDOUT.

TOUTMEM=*member*

The name of the template member in the dataset(s) identified by the TOUTPUT parameter, if it has not been specified on the DD statement. This parameter must not be specified if the TCOUT parameter is specified.

ROWS=*num*

The number of rows to be created. Valid range 1-99999999.

Example

Examples

Example 1: Create 1000 rows of test data in table "ID1"."TABLE1". The initialization information for each column of the table are defined in template EMPDG.

```
//D2G JOB (acct),'name'
//* Create rows in table ID2.TABLE1.
//*
```

```
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM D2G OBJOUT="ID1"."TABLE1",
$$FILEM TMOUT=ID1.HFM.TEMPLATE(EMPDG),
$$FILEM ROWS=1000
/*
```

Return codes

The return codes from the D2G function:

0

The function completed successfully

16

A serious error was encountered

Related functions

DBC

Copy data from one Db2® object to another (DBC (Copy) batch command on page 1027)

DBI

Import data from a sequential or VSAM file into a Db2® table (DBI (Import) batch command on page 1034)

DBX

Export data from a Db2® object to a sequential or VSAM file (DBX (Export) batch command on page 1043)

D2G

Create (generate) Db2® data (D2G (Data Generate) batch command on page 1058)

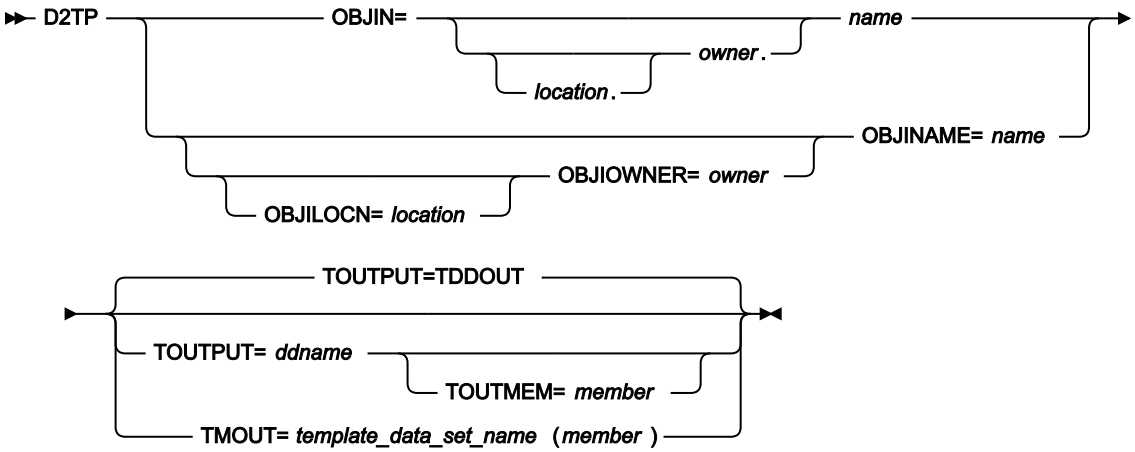
D2TP (Template create/update) batch command

Purpose

To create or update a template based on one Db2® object. The object must be accessible from the currently connected Db2® system.

Usage

The ZDT/Db2 template create/update utility creates a template if it does not exist, or updates the template if it does exist.



OBJIN location.owner.object

The optional name of the Db2® remote server (location) where the source object is located; the optional name of the owner of the source object (owner) and the source object name (name).

When location is not specified the current (local) Db2® server is used. When the owner is not specified the object name is qualified using the current SQLID. When ZDT/Db2 generates the utility control statements, the owner value is non-blank.

OBJIN should be used when the fully qualified name fits on a single line in the JCL deck. The last usable column is column 71. When the fully qualified name does not fit on a single line in the JCL deck, use one or more of the OBJILOCN, OBJIOWNR, OBJINAME keywords to specify the object.

OBJILOCN=*location*

The optional name of the Db2® remote server (*location*) where the source object is located. See Specifying a Db2 object name on page 1063.

OBJIOWNR=*owner*

The optional name of the owner of the source object (*owner*). See Specifying a Db2 object name on page 1063.

OBJINAME=*name*

The object name (*name*) for the source object. See Specifying a Db2 object name on page 1063.

TOUTPUT=*ddname*

Defines a reference to a DD statement for the data sets which contain the Db2® template that describes the target Db2® object. Concatenated DD statements are not supported and the referenced data set must be catalogued. If you have not specified a member name in the referenced DD statement, then you must provide a TOUTMEM keyword. If no TOUTPUT or TMOUT parameter has been provided, then TOUTPUT=TDDOUT is used. See Specifying the template for a Db2 object on page 1064.

TOUTMEM=*member*

The name of the template member in the dataset identified by the TOUTPUT or TMOUT parameter. This parameter is ignored if the member name is provided with the DD statement or the TMOUT parameter. See Specifying the template for a Db2 object on page 1064.

TMOUT=*template_data_set_name*(*member*)

The PDS (*template_data_set_name*) and member name (*member*) of the Z Data Tools template that describes the target Db2® object. See Specifying the template for a Db2 object on page 1064.

Example

Examples

Example 1: Create a Db2® template for the DSN8810.EMP using TMOUT keyword.

```
//D2TPJOB (acct),'name'
//* Create template HFM.TEMPLATE(EMP) for DSN8810.EMP table
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM D2TP OBJIN="DSN8810"."EMP",
$$FILEM TMOUT=HFM.TEMPLATE(EMP)
/*
```

Example 2: Use TOUTPUT, TOUTMEM keywords to create output template for DSN8810.EMP

```
//D2TPJOB (acct),'name'
//* Create template HFM.TEMPLATE(EMP) for DSN8810.EMP table
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//TOUT  DD DSN=HFM.TEMPLATE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM D2TP OBJIN="DSN8810"."EMP",
$$FILEM TOUTPUT=TOUT,TOUTMEM=EMP
/*
```

Example 3: Use Default output DD TDDOUT with TOUTMEM to create output template for DSN8810.EMP

```
//D2TPJOB (acct),'name'
//* Create template HFM.TEMPLATE(EMP) for DSN8810.EMP table
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
```

```
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//TDDOUT  DD DSN=HFM.TEMPLATE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM D2TP OBJIN="DSN8810"."EMP",
$$FILEM TOUTMEM=EMP
/*
```

Example 4: Use Default output DD TDDOUT with member to create output template for DSN8810.EMP

```
//D2TPJOB (acct),'name'
//* Create template HFM.TEMPLATE(EMP) for DSN8810.EMP table
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//TDDOUT  DD DSN=HFM.TEMPLATE(EMP),DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM D2TP OBJIN="DSN8810"."EMP"
/*
```

Example 5: Create 3 templates

```
//D2TPJOB (acct),'name'
//* Create template TOUT1-3 for DSN8810.EMP table
//*
//HFMDB2   EXEC PGM=HFMDB2,PARM=('SSID=DSN1,SQID=ID1')
//STEPLIB  DD DSN=HFM.SHFMMOD1,DISP=SHR
//         DD DSN=DB2V810.DSN1.SDSNEXIT,DISP=SHR
//         DD DSN=DB2.V810.SDSNLOAD,DISP=SHR
//TOUT1  DD DSN=HFM.TEMPLATE(T1),DISP=SHR
//TOUT2  DD DSN=HFM.TEMPLATE(T2),DISP=SHR
//TOUT3  DD DSN=HFM.TEMPLATE(T3),DISP=SHR
//SYSPRINT DD SYSOUT=*
//HFMTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM D2TP OBJIN="DSN8810"."EMP",TOUTPUT=TOUT1
$$FILEM D2TP OBJIN="DSN8810"."EMP",TOUTPUT=TOUT2
$$FILEM D2TP OBJIN="DSN8810"."EMP",TOUTPUT=TOUT3
/*
```

Specifying a Db2® object name

For those batch functions that require a Db2® object name, ZDT/Db2 automatically determines the appropriate keywords to specify the Db2® objects. For an input Db2® object this is OBJIN, or OBJILOCN (for location), OBJIOWNR (for owner) and OBJINAME (for name). For an output Db2® object this is OBJOUT, or OBJOLOCN (for location), OBJOOWNR (for owner) and OBJONAME (for name). The latter values are used when the OBJIN/OBJOUT value does not fit on a single line, or when the owner or name values contain special characters, for example, lowercase letters. In what follows, a special character is any

character other than the uppercase letters A-Z, the digits 0-9, the 3 characters reserved as alphabetic extenders for national languages (@#$), and the underscore character (_).

When the owner or name value is split over two or more lines, the rules for continuing Z Data Tools batch parameters must be observed. See the *Z Data Tools User's Guide* for a full explanation. Briefly, each part of the value must be enclosed in apostrophes, and a comma must appear at the end of each line. The second and subsequent lines should include $ $FILEM followed by a space and the next part of the parameter value, enclosed in apostrophes. When the owner/name value contains special characters ZDT/Db2 automatically determines a unique two-character delimiter for the parameter value. The default is \$, another delimiter is used if \$ appears in the owner/name. The two-character delimiter appears at the start and finish of the owner/name value. When the owner/name value spans two or more lines, the two-character delimiter appears at the start and end of every line, and apostrophes are used at the start and end of each partial parameter value - to satisfy the continuation rules for Z Data Tools batch parameters. When you code your own batch control statements and the owner/name values extend over 2 or more lines, you should specify each part of the name separately and not use OBJIN/ OBJOUT. If the owner/name values contain special characters, you must specify a two-character delimiter at the start and end of the parameter value (or partial parameter value). The delimiter must not appear in the owner/name value, and the first character of the delimiter must be a special character except for apostrophe and quotation mark characters. When 2 character delimiters are used for an owner/name value, specify the owner/name exactly as is, do not double apostrophes or quotation characters, or omit any spaces.

Specifying the template for a Db2® object

The Z Data Tools template for a Db2® object must be a Z Data Tools/Db2 template that was generated by ZDT/Db2. Unlike the Z Data Tools Base component, you cannot use a copybook to describe the structure of a Db2® object.

The JCL generated by ZDT/Db2 for the various batch functions uses the following batch keywords:

TMIN

For functions where there is an input Db2® object (export, print, copy, data create)

TMOUT

For functions where there is an output Db2® object (import, copy).

When it is necessary to specify the template data set name in the JCL statements, rather than via a Z Data Tools batch keyword, you can use the alternative keywords TINPUT and TINMEM.

Specifying the template/copybook for a data set

To describe the structure of a data set you can use:

- A COBOL copybook
- A HLASM copybook
- A PL/I copybook
- A Z Data Tools template

The JCL generated by ZDT/Db2 for the various batch functions always uses the TCIN or TCOUT batch keywords, as described below:

TCIN

For functions where there is an input data set (import)

TCOUT

For functions where there is an output data set (export)

When it is necessary to specify the template/copybook data set name in the JCL statements, rather than via a Z Data Tools batch keyword, you can use the alternative keywords TINPUT and TINMEM, or TOUTPUT and TOUTMEM.

Appendix A. Messages

In this section, selected batch messages are listed in alphanumeric order. For each message, the information provided comprises:

- The message identifier.
- The text of the message.
- An explanation of the message.
- The required user response.

Messages issued by ZDT/Db2 have a unique alphanumeric identifier with the format:

```
HFMDannn
```

where:

a

Is an alphabetic character.

nnn

Is a 3-digit number.

When the message is issued by a batch function, the identifier is printed in front of the text of the message.

When the message is issued by an online function, the identifier is not displayed with the message. However, you can obtain the identifier of a message by typing `MSGID` on the command line and pressing Enter. This command causes the message identifier of the last message issued to be displayed on your screen.

Many of the online messages comprise a short message and long message. The short message is displayed in the upper right of the panel. The long message is only displayed if you press the F1 key when the short message is displayed.

ZDT/Db2 messages

HFMDA001

HCL Z Data Tools Version 17 Release 0

Explanation: Shows the version and release information for Z Data Tools.

User response: No action required. Informational message.

HFMDA002

HCL Z Data Tools Db2® Component

Explanation: Shows the full component name for Z Data Tools.

User response: No action required. Informational message.

HFMDA003

HCL Z Data Tools/Db2

Explanation: Shows the abbreviated component name for Z Data Tools.

User response: No action required. Informational message.

HFMDA004

Db2® Component

Explanation: Shows the short component name for Z Data Tools.

User response: No action required. Informational message.

HFMDA009

HCL Z Data Tools Version 17 Release 0 Db2 Component
Licensed Materials - Property of IBM® and/or HCL
19OP1220
Copyright IBM® Corporation 2001, 2017 - All Rights Reserved. Copyright HCL Technologies Ltd. 2017, 2026 - All Rights Reserved. Trademark of International Business Machines

Explanation: Shows the product name and copyright information for Z Data Tools, Db2® component.

User response: No action required. Informational message.

HFMDA011

Panel *panel name* error. (short). PQUERY error processing ISPF panel *panel name*. (long)

Explanation: An ISPF error was encountered issuing a PQUERY service for panel name *panel name*.

User response: This is a Z Data Tools internal error. Contact HCL Support for assistance.

HFMDA012

Installation error. Db2® catalog table *Db2 catalog table name* is not available. Some ZDT/Db2 functions will be inoperative. Contact your Systems Programmer.

Explanation: ZDT/Db2 attempted to access the Db2® catalog table *Db2 catalog table name*. The access failed because *Db2 catalog table name* does not exist.

User response: Z Data Tools Db2® component requires access to various Db2® catalog tables for correct and optimal functionality. This message indicates an installation related problem. It is possible to define views of the Db2® catalog tables with a user-specified owner (example SYSIBMV), in place of the actual owner, (SYSIBM). One cause of this error is the use of a user-specified owner, but the views of the Db2® catalog tables with the user-specified owner do not exist (have not been created). Check the CATOWNER parameter, HFM2POPI macro in the HFM2POPT module. Check that views of the Db2® catalog tables, (for example SYSIBMV.SYSTABLES) exist. See *Preparing to customize ZDT/Db2* in the *Z Data Tools Customization Guide* for more information.

HFMDA013

Installation error. SELECT access to Db2® catalog table *Db2 catalog table name* is required, but is not available. Some ZDT/Db2 functions will be inoperative. Contact your Systems Programmer.

Explanation: ZDT/Db2 attempted to access the Db2® catalog table *Db2 catalog table name*. The access failed. Access to *Db2 catalog table name* is essential to complete the current ZDT/Db2 function.

User response: ZDT/Db2 requires SELECT access to various Db2® catalog tables for correct and optimal functionality. This message indicates an installation problem; specifically that the Db2® authid used by the ZDT/Db2 user does not have SELECT access to the Db2® catalog table *Db2 catalog table name*. See *Preparing to customize ZDT/Db2* in the *Z Data Tools Customization Guide*, and to point 8 in "Grant access to the Db2® catalogs", for more information. SELECT access to *Db2 catalog table name* needs to be GRANTed to the ZDT/Db2 user who encountered the error.

HFMDA014

Installation error. Access to column *column name* in Db2® catalog table *Db2 catalog table name* is required, but is not available. Some ZDT/Db2 functions will be inoperative. Contact your Systems Programmer.

Explanation: ZDT/Db2 attempted to access column *column name* of Db2® catalog table *Db2 catalog table name*. The access failed. SELECT access to column *column name* of *Db2 catalog table name* is essential to complete the current ZDT/Db2 function.

User response: ZDT/Db2 requires access to various Db2® catalog tables for correct and optimal functionality. This message indicates an installation related problem, specifically that the Db2® authid used by the ZDT/Db2 user does not have SELECT access on column *column name* in Db2® catalog table *Db2 catalog table name*. *Preparing to customize ZDT/Db2* in the *Z Data Tools Customization Guide*, and to point 8 in the checklist "Grant access to the Db2® catalogs - required". SELECT access to column *column name* of *Db2 catalog table name* needs to be GRANTed to the ZDT/Db2 user who encountered the error.

HFMDA020

REQUIRED, BROWSE AUDITED

Explanation: Auditing is required, browse functions are subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA021

REQUIRED, BROWSE NOT AUDITED

Explanation: Auditing is required, browse functions are not subject to audit. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA022

REQUIRED, SMF, BROWSE AUDITED

Explanation: Auditing is required, and auditing records are written to SMF, browse functions are subject to audit. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA023

REQUIRED, SMF, BROWSE NOT AUDITED

Explanation: Auditing is required, and auditing records are written to SMF, browse functions are not subject to audit. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA024

None

Explanation: ZDT/Db2 component does not write audit records. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA025

SAF-RULE CONTROLLED

Explanation: Auditing within ZDT/Db2 is controlled using SAF rules, defined using a security product such as RACF®. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA026

SAF-RULE CONTROLLED, AUDITING NOT ALLOWED

Explanation: Auditing within ZDT/Db2 is controlled using SAF rules, defined using a security product such as RACF®. These rules specifically prevent the writing of audit records by ZDT/Db2. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA027

DEMAND, BROWSE AUDITED

Explanation: Demand auditing is in effect, browse functions are subject to audit. Demand auditing means that an audit report is automatically printed at the end of the current editor session. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the"Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA028

DEMAND, BROWSE NOT AUDITED

Explanation: Demand auditing is in effect, browse functions are not subject to audit. Demand auditing means that an audit report is automatically printed at the end of the current editor session. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA030

OPTIONAL, OFF, BROWSE AUDITED

Explanation: Auditing is optional and initially off, browse functions are subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the"Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA031

OPTIONAL, OFF, BROWSE NOT AUDITED

Explanation: Auditing is optional and initially off, browse functions are not subject to audit. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA032

OPTIONAL, PROFILE, BROWSE AUDITED

Explanation: Auditing is optional and the current value is set based on the user ISPF profile. Browse functions are subject to audit. Auditing options are specified in either the HFM2POPT module or externally, using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA033

OPTIONAL, PROFILE, BROWSE NOT AUDITED

Explanation: Auditing is optional and the current value is set based on the user ISPF profile. Browse functions are not subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as

RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA034

OPTIONAL, SMF, OFF, BROWSE AUDITED

Explanation: Auditing is optional and audit records are written to SMF. The audit option is initially off, and browse functions are subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA035

OPTIONAL, SMF, OFF, BROWSE NOT AUDITED

Explanation: Auditing is optional and audit records are written to SMF. The audit option is initially off, browse functions are not subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA036

OPTIONAL, SMF, PROFILE, BROWSE AUDITED

Explanation: Auditing is optional and audit records are written to SMF. The current value for the audit option is set based on the user ISPF profile, browse functions are subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA037

OPTIONAL, SMF, PROFILE, BROWSE NOT AUDITED

Explanation: Auditing is optional and audit records are written to SMF. The current value for the audit option is set based on the user ISPF profile, browse functions are not subject to audit. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA038

DEMAND, SMF, BROWSE AUDITED

Explanation: Demand auditing is in effect, audit records are written to SMF, browse functions are subject to audit. Demand auditing means that an audit report is automatically printed at the end of the current editor session. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA039

DEMAND, SMF, BROWSE NOT AUDITED

Explanation: Demand auditing is in effect, audit records are written to SMF, browse functions are not subject to audit. Demand auditing means that an audit report is automatically printed at the end of the current editor session. Auditing options are specified in either the HFM2POPT module, or externally using a security product such as RACF®. See *Z Data Tools Customization Guide* for more information. This message is displayed on the "Db2® connection Information" pop-up panel.

User response: No action required.

HFMDA040

Recovered from Db2® abend (short). ZDT/Db2 has recovered from an abend in Db2®. The ZDT/Db2 function that was executing when the Db2® abend occurred has been terminated. (long)

Explanation: A Db2® abend occurred during processing for the current function. The Z Data Tools subtask responsible for connection to Db2® has been terminated, and a new subtask established. Any uncommitted changes made by the ZDT/Db2 function prior to the abend will have been backed out.

User response: Document information about the function being used when the abend occurs. If the problem occurs again notify your System Programmer. If there is no Db2-related problem that might explain the abend, contact HCL Support.

HFMDA050

Object not found (*ID*) (Short). No Db2® object of the appropriate type matches the combination of object details entered in Db2® system *ID*. Press Help for more information (Long).

Explanation: This message is issued when a Db2® object name has been entered on a function entry panel, but Db2® reports that the object name does not exist. Db2® SSID is the name of the currently connected Db2® sub-system. The function cannot be processed until the reason for the error is determined and corrected.

User response:

Press the the Help Pfkey (PF1 by default) twice to show, in order, the long message (see above) and the extended help, which is a panel. The information on the help panel is shown below.

- The specified Db2® object does not exist:
  - Is the spelling correct?
  - Are you connected to the correct Db2® system?
- There was an uppercase/lowercase mismatch. The Translate Db2® object name option is OFF, if the name that was entered contains lower case characters, it may not be recognized. To change the Translate Db2® object name option select option 0.0.2.
- If no OWNER was specified, ZDT/Db2 used *authid* to qualify the name.

- A synonym name was specified, but the owner of the synonym is not *authid*. A synonym name is only recognized when the current SQLID and the synonym owner match.
- The object name specified is too long to be displayed on the function entry panel. If you used EXPAND to enter a long name in a field, you must use EXPAND again to clear all parts of the name. Clearing the field on the function entry panel only sets the visible part of the field to spaces.

Read the information on the panel and consider the various causes of the error in turn until the cause is determined.

HFMDA051

Invalid location (Short). Location *LOC* is not defined in SYSIBM.LOCATIONS at the local server (Long).

Explanation: This message is issued when a Db2® location value has been entered on a function entry panel, but Db2® reports that the location cannot be found. *Location* is the name of the location value that could not be located. The function cannot be processed until the reason for the error is determined and corrected.

User response: You can enter * in the location field and press Enter to show a list of locations defined at the local server. This feature may not be available if the product installer has chosen to disable it. If you cannot determine the reason why the entered location value is invalid contact your Db2® administrator for assistance.

HFMDA052

Object not found (Short). No Db2® object at location *LOC* matches the combination of object details entered at Db2® system *ID*. Press Help for more information. Object not found (short).

Explanation: This message is issued when a Db2® object name has been entered on a function entry panel, but Db2® reports that the object name does not exist. *Db2 SSID* is the name of the currently connected Db2® sub-system. *Location* is the value entered in the LOCATION entry field. The function cannot be processed until the reason for the error is determined and corrected.

User response: Press the Help PFkey (PF1 by default) twice to show, in order, the long message (see above) and the extended help, which is a panel. The information on the help panel is shown below: Possible reasons why Db2® could not find this object include:

- The specified Db2® object does not exist:
  - Is the spelling correct?
  - Have you specified the correct location?
- There was an uppercase/lowercase mismatch. The Translate Db2® object name option is OFF, if the name that was entered contains lower case characters, it may not be recognized. To change the Translate Db2® object name option select option 0.0.2.
- If no OWNER was specified, ZDT/Db2 used *authid* to qualify the name.
- A synonym name was specified, but the owner of the synonym is not specified. A synonym name is only recognized when the current SQLID and the synonym owner match.
- The object name specified is too long to be displayed on the function entry panel. If you used EXPAND to enter a long name in a field, you must use EXPAND again to clear all parts of the name. Clearing the field on the function entry panel only sets the visible part of the field to spaces.

Read the information on the panel and consider the various causes of the error in turn until the cause is determined.

HFMDA053

Check name (spaces) (Short). The Db2® object name cannot be found. There are two or more consecutive spaces in the object name, this might indicate unwanted text in the non-visible parts of a scrollable field. Place the cursor on the object name field and press the Expand PF key to verify the contents. (long).

Explanation: A value entered in a name field has two or more consecutive spaces, for example: `ABC  FG`. The name field is an ISPF scrollable field, allowing up to 128 characters to be entered, however not all characters are displayed. It is possible that unwanted information has been entered in the part of the field that is not displayed. Because ZDT/Db2 validates the DB2® name based on the entire contents of the field, *"Object not found"* errors will occur.

Explanation: This message is issued when a Db2® object name has been entered on a function entry panel, but Db2® reports that the object name does not exist. The Db2® object name contains two or more consecutive spaces, which may be valid, but more likely indicates the cause of the error.

User response: Place the cursor on the object name field and press the PFKey assigned to EXPAND (PF4 by default) to show the complete field contents. If the EXPAND key is not available you can type the EXPAND command on the command line, position the cursor on the object name field and press ENTER to show the complete field contents. Remove any spurious data and retry the operation.

HFMDA054

Check owner (spaces) (Short). The Db2® object name cannot be found. There are two or more consecutive spaces in the object owner, this might indicate unwanted text in the non-visible parts of a scrollable field. Place the cursor on the object owner field and press the Expand PF key to verify the contents.

Explanation: A value entered in an owner field has two or more consecutive spaces, for example: `ABC  FG`. The owner field is an ISPF scrollable field, allowing up to 128 characters to be entered, however not all characters are displayed. It is possible that unwanted information has been entered in the part of the field that is not displayed. Because ZDT/Db2 validates the DB2® owner based on the entire contents of the field, *"Object not found"* errors will occur.

User response: Place the cursor on the object owner field and press the PFKey assigned to EXPAND (PF4 by default) to show the complete field contents. If the EXPAND key is not available you can type the EXPAND command on the command line, position the cursor on the object owner field and press ENTER to show the complete field contents. Remove any spurious data and retry the operation.

HFMDA055

No Db2® TABLE matches the value entered. It is possible a Db2® object (eg view) that matches the name entered exists - however the selected Db2® utility supports Db2® tables ONLY. (long).

Explanation: This message is issued by the Db2® utilities function when an object not found condition is encountered. Some Db2® utilities require a table as input, and will not process a view, synonym or other objects.

User response: Use the ZDT/Db2 object list utility to display information about the Db2® object. Determine whether the object exists, and determine the type of the object (table, view etc). Consult the Db2® utilities manual for the appropriate version of Db2® to determine which object types may be used.

HFMDA100

Alloc. error -*unit* (short).The attempt to allocate a temporary data set for the SQL statement failed. UNIT *unit* is invalid. Check the temporary data set allocation option values. (long).

Explanation: Z Data Tools attempted to allocate a temporary data set, in which to store SQL or some other information. The allocation attempt failed, the most likely explanation is that the UNIT parameter specified on the "Set Temporary Data Set Allocation Options" panel is invalid. This system option panel is accessed using 0.0.8 from the ZDT/Db2 main menu panel.

User response: Check the value specified for the UNIT parameter on the "Set Temporary Data Set Allocation Options" panel. Remove any value and retry the operation. If this resolves the problem, contact your system's programmer for advice on valid/invalid UNIT values.

HFMDA997

**** Limit of Rows Retrieved ****

Explanation: This message appears at the bottom of a Z Data Tools/Db2 editor session, when a row count value (example 100) was specified on the editor function entry panel, and ZDT/Db2 read that many rows into the editor session without reaching the end of the Db2® object.

User response: No action required. Informational message.

HFMDA998

Internal Logic Error (*module name*) - *message*

Explanation: This is an internal error message, indicating a problem with the processing of the Db2® attachment.

User response: Contact HCL Support.

HFMDB000

JESCT not available (short). JESCT control block currently not available for Db2® subsystem selection. (long).

Explanation: This is an internal error. ZDT/Db2 attempted to access the JESCT however the pointer was zero.

User response: Contact HCL Support.

HFMDB001

No Db2® subsystems (short). No Db2® Subsystems have been defined on this system. Z Data Tools Db2 component can not run on this system. (long).

Explanation: ZDT/Db2 could not find any Db2® systems defined on the current z/OS® system.

User response: If there are Db2® systems defined on the current z/OS® system, contact HCL Support.

HFMDB002

SSID *Db2 SSID* not matched (short). No Db2® Subsystems matching *Db2 SSID* are defined on this system. (long).

Explanation: This message may be issued during Db2® subsystem selection when a name with a wildcard (example DSN*) is entered in the Db2® SSID field on the ZDT/Db2 main menu. No matching Db2® systems were found.

User response: No action required. Informational message. Clear the Db2® SSID entirely to show all Db2® systems.

HFMDB003

Db2 SSID not defined to z/OS® (short) . Db2® Subsystem/Group *Db2 SSID* is not defined to the local z/OS® system. (long).

Explanation: Db2® system *Db2 SSID* is not defined on the local z/OS® system. This message may be issued in response to an attempt to change the currently connected Db2® system using a command like SSID XXXX.

User response: Check that *Db2 SSID* is correctly specified. Correct the name and try again.

HFMDB004

SSID *Db2 SSID* not active (short). Db2® subsystem/group *Db2 SSID* is not currently active. (long).

Explanation: The Db2® system or group *Db2 SSID* is defined on the local z/OS® system but is inactive (stopped).

User response: ZDT/Db2 cannot connect to inactive Db2® systems. Determine the reason why the Db2® system is inactive. When the Db2® system is started, attempt the connection again.

HFMDB005

Already connected (short). ZDT/Db2 is currently connected to Db2® system *Db2 SSID*. (long).

Explanation: ZDT/Db2 is already connected to Db2® system *Db2 SSID*. This message may be issued in response to an attempt to change the currently connected Db2® system using a command like SSID XXXX.

User response: None if *Db2 SSID* was correctly specified. Otherwise, check the Db2® SSID, make appropriate corrections, and try again.

HFMDB006

Cannot connect to Db2® *Db2 SSID*, RRSAF *function* RC &RC. Reason *reason code*.

Explanation: A call to the Db2® RRSAF attachment failed. *Db2 SSID* is the currently connected Db2® system; *function* is the failing RRSAF function and *reason* is the failing reason code.

User response: This error indicates an installation error. Confirm that the RRSAF attachment is required. You will typically find that the CAF attachment meets all your requirements. The Db2® attachment used by ZDT/Db2 is specified in the HFM2POPT module. See *Z Data Tools Customization Guide* for a detailed explanation. Ensure that the appropriate Db2® interface module (DSNRLI) is accessible to ZDT/Db2. If these actions fail to resolve the problem, contact HCL Support.

HFMDB007

A Db2® Subsystem has not been specified.

Explanation: A Db2® system has not been specified. ZDT/Db2 must connect to a Db2® system before it can perform useful work.

User response: Use the "Db2® SSID" field on the ZDT/Db2 main menu to specify the name of a Db2® system.

HFMDB008

Subsystem *Db2 SSID* does not identify a uniquely active Db2® subsystem

Explanation: A Db2® SSID value was specified, but it does not uniquely identify a Db2® system. An example is DSN*.

User response: Specify an unambiguous value.

HFMDB009

Db2® subsystem *Db2 SSID* is not currently active on this system.

Explanation: The specified Db2® SSID value *Db2 SSID* is inactive.

User response: Take action to start the Db2® system. When the Db2® system is active, retry the operation.

HFMDB010

Row *Row value 1* of *Row value 2*

Explanation: This message is displayed on various selection lists, including the Db2® subsystem selection panel. *Row value 1* is the first row shown on the display and there are a total of *Row value 2* rows available for display.

User response: No action required. Informational message.

HFMDB011

Invalid selection (short). Enter "S" or "/ " to select the required Db2® subsystem. (long).

Explanation: A character other than "S" or "/ " was entered against an entry on the Db2® subsystem selection panel. The only valid characters for selecting an entry are "S" or "/ ".

User response: Use either "S" or "/ " to select the required entry.

HFMDB012

SSID *Db2 SSID* not active (short). Selected Db2® subsystem *Db2 SSID* is not currently active on this system. (long).

Explanation: An attempt was made to select Db2® system *Db2 SSID* from the Db2® subsystem selection panel. This Db2® system is not currently active.

User response: Determine why the Db2® system is inactive and take action to start it. When the Db2® system is active, retry the operation.

HFMDB013

Selection terminated (short) Db2® subsystem selection was terminated with the EXIT *PF key* command. Z Data Tools Db2® is not currently connected to any Db2® subsystem. (long)

Explanation: The user pressed the EXIT *PF key* on the Db2® subsystem selection menu without making any selection. The ZDT/Db2 main menu is re-displayed. ZDT/Db2 is not connected to any Db2® system.

User response: No action required. Informational message. It is necessary to connect to a Db2® system before performing any ZDT/Db2 functions.

HFMDB014

Display refreshed (short). List of Db2® Subsystems has been refreshed using current information. (long).

Explanation: The user issued the REFRESH primary command on the Db2® subsystem selection list panel. The list of Db2® systems displayed is refreshed.

User response: No action required. Informational message.

HFMDB015

SSID *Db2 SSID* displayed (short). Db2® Subsystems matching *Db2 SSID* are displayed. (long).

Explanation: The user issued a SHOW *Db2 SSID* command on the Db2® subsystem selection list panel. The list of Db2® systems that match the SSID specification are displayed. The SSID value specified may include wild cards, example DSN*.

User response: No action required. Informational message.

HFMDB016

Command not active (short). The command to change the Db2® subsystem is not available from this panel. (long).

Explanation: A command to change the currently connected Db2® SSID was issued on a panel where the command is not available.

User response: The currently connected Db2® SSID can only be changed when no ZDT/Db2 function is active. Return to a function-entry, or menu panel, and reissue the command.

HFMDB017

SSID name required (short). Enter the Db2® subsystem name that ZDT/Db2 should connect to. (long).

Explanation: A command to change the currently connected Db2® system was issued without an operand.

User response: The command to change the currently connected Db2® system requires an operand, the Db2® SSID of the Db2® system to which connection is desired. Reissue the command with the required Db2® system SSID as the operand.

HFMDB018

Invalid remote server (short). Remote server *Db2 server name* is not a Db2® UDB for z/OS® system. Connection to remote servers that are not Db2® UDB for z/OS® servers is not supported. (long).

Explanation: A Db2® server name was entered in the location field on a ZDT/Db2 function entry panel. While the Db2® server name is defined as a remote Db2® server at the local Db2® system, this server is not a Db2® UDB for z/OS® system. ZDT/Db2 only supports remote connections to Db2® UDB for z/OS® servers.

User response: Connection to this remote Db2® server is not supported. Specify another name.

HFMDB019

Version *Db2 version* invalid (short). The remote Db2® version/release is *Db2 version*. Connection is supported for Db2® Version 9 and later systems only. (long).

Explanation: A Db2® server name was entered in the location field on a ZDT/Db2 function entry panel. The version of the remote Db2® system is *Db2 version*. Connection to this version of Db2® is not supported.

User response: Connection to this remote Db2® server is not supported. Specify another name.

HFMDB020

Operating System is unsupported. Can not run Z Data Tools Db2 component.

Explanation: This is an internal error. During MVS™ token initialization an unsupported version of z/OS® was detected. It is not possible to run ZDT/Db2 against this version of z/OS®, or its predecessors.

User response: Ensure that the host operating system is at a level specified in the Program Directory. If you believe that the host operating system is at a supported level, contact HCL Support.

HFMDB021

Error in *function* Callable Service Routine, Return code *return code*.

Explanation: This is an internal error. A call to *function* failed with return code *return code*.

User response: Contact HCL Support.

HFMDB022

Db2® Subsystem not set (short). Error invoking Z Data Tools Db2 component. Db2® Subsystem has not been set. (long).

Explanation: An attempt was made to run ZDT/Db2, without correctly specifying a Db2® SSID name, or without first connecting to a Db2® system.

User response: If the message occurs in batch, check that the Db2® SSID value is correctly specified in the JCL. If this fails to resolve the problem, start ZDT/Db2 online and regenerate the batch JCL. If the batch job still fails contact HCL Support. If the message occurs online, return to the ZDT/Db2 main menu and enter the name of a valid Db2® SSID in the Db2® SSID field, press Enter to connect.

HFMDB023

Cannot connect to Db2® *Db2 SSID*, CAF *function* RC *return code*, Reason *reason code*.

Explanation: ZDT/Db2 attempted to call the Db2® CAF interface with function *function*. The call failed with return code *return code* and reason code *reason code*.

User response: This message usually indicates an installation or set up problem. The most likely causes are:

1. Not binding the ZDT/Db2 plan (HFM2PLAN) by default.
2. Specifying an incorrect Db2® plan name in the HFM2POPT module.

See *Z Data Tools Customization Guide* for more information.

HFMDB024

Unable to load Db2® Interface module *module name*. Abend Code *abend code-reason code*. Check STEPLIB/LINKLIST for Db2® load libraries.

Explanation: ZDT/Db2 attempted to load the Db2® interface module *module name*. The load failed with abend code *abend code* and reason *reason code*.

User response: This is probably a customization or setup problem. If the Abend is 806 (module not found):

1. Check that the Db2® load library has been correctly specified in the HFM2SSDM macro for the Db2® system in question.
2. If the HFM2POPT module does not include the Db2® load library names for the Db2® system, check any applicable STEPLIB/JOBLIB and LINKLIST for the Db2® load library. The Db2® load library must reside in a library that is accessible using the standard search order for the LOAD macro.

For other abends refer to point (1) as a starting point. If this fails to resolve the problem, contact HCL Support.

HFMDB025

Not currently connected to a Db2® subsystem.

Explanation: ZDT/Db2 is not currently connected to any Db2® subsystem.

User response: No action required. Informational message. It will be necessary to connect to a Db2® system prior to performing any ZDT/Db2 functions.

HFMDB026

Subsystem: *Db2 SSID*, Db2® release: *Db2 VRM indicator*.

Explanation: ZDT/Db2 is connected to Db2® system *Db2 SSID*. The version of this Db2® system is *Db2 VRM indicator*.

User response: No action required. Informational message. issued in a Z Data Tools/Db2 batch job.

HFMDB027

ZDT/Db2 plans: *Plan name 1*, *Plan name 2*

Explanation: The ZDT/Db2 plan names are *Plan name 1* and *Plan name 2*.

User response: No action required. Informational message. issued in a Z Data Tools/Db2 batch job.

HFMDB028

Description: *user specified description*.

Explanation: This message *user specified description* appears in the" About Db2®" popup panel. The *user specified description* is the value specified in the HFM2SSDM macro, DESC parameter for the appropriate Db2® system.

User response: No action required. Informational message.

HFMDB029

Db2 SSID executing Db2® level *Db2 level 1*. Minimum supported level is *Db2 level 2*.

Explanation: This message is issued when an attempt is made to connect to a Db2® system at an unsupported version/maintenance level. The level of Db2® system *Db2 SSID* is *Db2 level 1*. The minimum supported level is *Db2 level 2*.

User response: Connection to the Db2® system is not supported. See *Z Data Tools Customization Guide* and *Z Data Tools User's Guide and Reference* for information on supported Db2® versions, for the version of ZDT/Db2 in use.

HFMDB030

Db2I unavailable (short). Unable to start Db2I. Check STEPLIB/LINKLIST for Db2® libraries. (long).

Explanation: A ISPF PQUERY service for panel DSNEPRI was issued, the request failed.

User response: This error indicates an installation or setup problem. Ensure that the Db2® libraries are available to the ZDT/Db2 user. The recommended way to do this is to code the Db2® libraries in the HFM2SSDM macro entry for each Db2® system that will be accessed by ZDT/Db2. See *Z Data Tools Customization Guide* for more information.

HFMDB031

SQL ID not authorized (short). You are not authorized to use the SQL user ID *Db2 authid* on Db2® Subsystem *Db2 SSID*. (long).

Explanation: An attempt was made to change the current SQLID on Db2® system *Db2 SSID* to *Db2 authid*. The attempt failed. The usual reason for this is that the ZDT/Db2 user is not authorized to use *Db2 authid*.

User response: Consult with either the Db2® or Security Administrator as to the reason why access to *Db2 authid* is not available.

HFMDB032

You are not authorized to use the SQL user ID *Db2 authid.*

Explanation: An attempt was made to change the current SQLID to *Db2 authid*. The attempt failed. The usual reason for this is that the ZDT/Db2 user is not authorized to use *Db2 authid*.

User response: Consult with either the Db2® or Security Administrator as to the reason why access to *Db2 authid* is not available.

HFMDB033

Auditing: *Audit status* :

Explanation: This message is used to display the audit status for the currently connected Db2® system, and appears on the "About Db2®" pop-up panel.

User response: No action required. Informational message.

HFMDB034

No Db2® subsystem with name *Db2 SSID*. CAF *function* RC *return code*, Reason *reason code*.

Explanation: A call to the CAF interface, function *function* for Db2® system *Db2 SSID* failed with RC *return code* and reason *reason code*. The reason is that the nominated Db2® system does not exist.

User response: Check the Db2® system name and try again.

HFMDB035

Db2® group/subsystem *Db2 SSID* is not active, CAF *function* RC *return code*, Reason *reason code*.

Explanation: A call to the CAF interface, function *function* for Db2® system *Db2 SSID* failed with RC *reason code* and reason *reason code*. The reason is that the nominated Db2® system is not active.

User response: Start the Db2® system. Retry the operation.

HFMDB036

Not authorized to connect to *Db2 SSID*, CAF function *function* RC *return code*, Reason *reason code*.

Explanation: A call to the CAF interface, function *function* for Db2® system *Db2 SSID* failed with RC *return code* and reason *reason code*. The likely reason is that the ZDT/Db2 user is not authorized to connect to the nominated Db2® system.

User response: Consult with either the Db2® or Security Administrator as to the reason why the ZDT/Db2 user is not authorized to connect to the Db2® system *Db2 SSID*.

HFMDB037

ZDT/Db2 not installed on *Db2 SSID*, CAF function *function* RC *return code*, Reason*reason code*.

Explanation: A call to the CAF interface, function *function* for Db2® system *Db2 SSID* failed with RC *return code* and reason *reason code*. The reason is that ZDT/Db2 has not been installed on Db2® system *Db2 SSID*.

User response: Verify that ZDT/Db2 is installed on Db2® system *Db2 SSID*. Retry the operation.

HFMDB038

Unable to load options module *module name*. Probable installation error.

Explanation: ZDT/Db2 attempted to load module *module name*. The attempt failed. The cause is an installation error; the Z Data Tools load library is not available to the ZDT/Db2 user.

User response: Check the allocation of the ZDT/Db2 load library using, for example, 'TSO ISRDDN&' (under ISPF). Ensure that *module name* is located in a load library accessible to the ZDT/Db2 user.

HFMDB039

HFM2POPI module invalid (short). The internal structure of HFM2POPT is invalid. Probable installation error. (long).

Explanation: ZDT/Db2 loaded the HFM2POPT module, the contents are corrupt and unusable.

User response: The HFM2POPT module was assembled using an older version of ZDT/Db2. Reassemble the HFM2POPT module, paying attention to the SYSLIB statements in the JCL, to ensure the correct versions of the ZDT/Db2 macros are used. Retry the operation.

HFMDB040

SSID *Db2 SSID* not available (short). Installation settings prevent the selection of Db2® subsystem/group *Db2 SSID*. (long).

Explanation: An attempt was made to connect to Db2® system *Db2 SSID*. This Db2® system is marked "Unavailable" in the HFM2POPT installation module. The Db2® system cannot be used.

User response: Select another Db2® system.

HFMDB041

Incompatible connection already active, CAF *function*, RC *return code*, Reason*reason code*.

Explanation: ZDT/Db2 attempted to call the CAF interface, function *function*. An error occurred, RC *return code* and Reason *reason code*. The cause is an incompatibility (release level mismatch) between the Db2® and call attachment code.

User response: This is an installation error. Check the specification of the Db2® libraries in the HFM2POPT, HFM2SSDM macro for the appropriate Db2® system, with attention to the Version/maintenance level of these libraries compared with the Version/maintenance level of the Db2® system.

HFMDB042

Db2 SSID Release Minimum Release (short). *Db2 SSID* executing Db2® level *Release*. The minimum supported level of Db2® for this release of Z Data Tools is *Minimum Release*. ZDT/Db2 may operate successfully with Db2® Version 8, however no support will be provided for any Version 8 specific problems that may be encountered.

Explanation: ZDT/Db2 connected to *Db2 SSID* which is at version and release *Release*. The minimum supported Db2® version/release for this version of ZDT/Db2 is *Minimum Release*, which is greater than *Release*.

User response: No action required. Informational message. ZDT/Db2 should operate without issues. However should a problem be encountered that is specific to Db2® version 8, no support will be available.

HFMDB043

Db2 SSIDReleaseMaximum Release (short). *Db2 SSID* executing Db2® level *Release*. This version and release of Z Data Tools does not currently support connection to Db2® *Maximum Release* (or later) subsystems.

Explanation: An attempt was made to connect to *Db2 SSID* which is at version and release *Release*. The maximum supported Db2® version/release for this version of ZDT/Db2 is *Maximum Release*, which is less than *Release*. The connection is not established and the Db2® system cannot be accessed. All versions of software are designed to work with a limited set (of version and release levels) of other software.

User response: Contact HCL Support for information about more recent versions of ZDT/Db2 which may support the Db2® version/release of *Db2 SSID*.

HFMDB044

Communications error (short). ZDT/Db2 encountered an SQL error *SQL error code* attempting to access the Db2® system at *Db2 location name*. This indicates that the remote Db2® system is not available, or that there was a security related problem. (long).

Explanation: ZDT/Db2 attempted to connect to remote Db2® server *Db2 location name*. The connection failed with SQLCODE *SQL error code*.

User response: Look up *SQL error code* in the Db2® messages and codes manual and take the action indicated. This error indicates a set up problem. One possibility is incorrect DRDA® definitions for Db2® location name at the local Db2® server.

HFMDB045

Not defined.

Explanation: A location nickname was not defined in the HFM2POPT module.

User response: Check the HFM2SSDM definitions in the HFM2POPT module and add the location nickname to the appropriate entry.

HFMDB046

Defined but no location specified.

Explanation: A location nickname was specified, but no location, to which the nickname refers, was specified.

User response: Check the HFM2SSDM macro for the currently connected Db2® system. Ensure that both a location and location nickname are specified.

HFMDB047

Authorization error (short). ZDT/Db2 encountered an SQL error (-725) attempting to access the Db2® system at *Remote location name*. This can occur when the user has changed the SQLID at the local Db2® system, but the user does not have authority to use this SQLID at the remote Db2® system. (long).

Explanation: Refer to the long message.

User response: Ensure that the user has access to the remote Db2® system *Remote location name* using the currently set Db2® SQLID.

HFMDB048

HFM2DATA load failed (short). Unable to load HFM2DATA module. Probable installation error. (long).

Explanation: ZDT/Db2 attempted to load data module HFM2DATA. The attempt failed. ZDT/Db2 cannot operate without this module.

User response: This is a customization or set up problem. Check that HFM2DATA is present in the load libraries accessible to the ZDT/Db2 user. This module is normally located in the SHFMMOD1 library.

HFMDB049

HFM2DENU load failed (short). Unable to load HFM2DENU module. Probable installation error. (long).

Explanation: ZDT/Db2 attempted to load data module HFM2DENU. The attempt failed. ZDT/Db2 cannot operate without this module

User response: This is a customization or setup problem. Check that HFM2DENU is present in the load libraries accessible to the ZDT/Db2 user. This module is normally located in the SHFMMOD1 library.

HFMDB050

ATTACH of ZDT/Db2 subtask program failed with return code *return code*.

Explanation: ZDT/Db2 attempted to attach the subtask responsible for Db2® access. The attempt failed with RC *return code*. ZDT/Db2 cannot operate without the subtask for consistency.

User response: This is an internal error. Retry the operation. If the problem persists check for installation related problems and contact HCL Support.

HFMDB051

Communications error (SQLCODE=*SQL error code*) accessing *Db2 SSID*.

Explanation: ZDT/Db2 attempted to access *Db2 SSID* and received a communications error *SQL error code*.

User response: Look up SQL error code in the Db2® messages and codes manual and take the action indicated.

HFMDB052

Db2® server *server name* is not a Db2® UDB for z/OS® server, code=*return code*.

Explanation: ZDT/Db2 (in batch) connected to Db2® server *server name*. The connect information indicates that the Db2® server is not a Db2® UDB for z/OS® server. Connection is only supported to Db2® UDB for z/OS® servers.

User response: Specify the name of a Db2® UDB for z/OS® server only.

HFMDB053

SET DECFLOAT ROUNDING MODE failed with return code *return code*.

Explanation: ZDT/Db2 attempted to set the DECFLOAT ROUNDING MODE, the attempt failed with return code *return code*.

User response: Check that the Db2® system is operational. If the problem persists contact HCL Support.

HFMDB054

SSID Db2® SSID not defined (short). There is no definition (HFM2SSDM macro entry) for Db2® subsystem or group *Db2 SSID* in the ZDT/Db2 installation module (HFM2POPT). Contact the person who installed ZDT/Db2. (long).

Explanation: The HFM2POPT installation module is used to specify the Db2® systems that ZDT/Db2 can connect to. The *Z Data Tools Customization Guide* recommends defining an HFM2SSDM macro entry for every Db2® system that ZDT/Db2 will connect to. The HFM2POPI macro includes a CONNECT option, the default setting is DEFINED. When CONNECT=DEFINED is in effect, ZDT/Db2 will only attempt connection to Db2® systems that have an HFM2SSDM entry in the HFM2POPT. An attempt to connect to a Db2® system with no HFM2SSDM entry results in this error.

User response: There are two usual approaches for resolving this error:

1. Add an HFM2SSDM macro entry for every Db2® system that ZDT/Db2 will access to the HFM2POPT, as recommended in the *Z Data Tools Customization Guide*.
2. Specify CONNECT=ANY in the HFM2POPI macro in the HFM2POPT. While this will allow ZDT/Db2 to attempt connection to ANY Db2® system, it does not necessarily remove the need to define an HFM2SSDM macro entry for every Db2® system that ZDT/Db2 will connect to. See the *Z Data Tools Customization Guide* for detailed setup information.

HFMDB055

Selection canceled (short). Db2® subsystem selection was canceled. Connection to the previously connected Db2® system has been re-established.

Explanation: The user exited the Db2® subsystem selection panel without selecting a Db2® subsystem or group

User response: No action required. This is an informational message.

HFMDB056

Selection canceled (short). Db2® subsystem selection was canceled. Z Data Tools Db2® is not currently connected to any Db2® subsystem.

Explanation: The user exited the Db2® subsystem selection panel without selecting a Db2® subsystem or group. ZDT/Db2 is not currently connected to any Db2® subsystem or group.

User response: No action required. This is an informational message. It will be necessary to connect to a Db2® system prior to performing any ZDT/Db2 functions.

HFMDB057

Db2® limits the output returned from a Db2® command. This limit has been reached and the output has been truncated.

Explanation: A Db2® command that produced large amounts of output was entered. Db2® has limited the amount of output produced, resulting in truncation.

User response: No action required. This is an informational message. Refine the Db2® command to be more specific.

HFMDB058

You cannot change the SQLID until ZDT/Db2 is connected to a Db2® subsystem or group.

Explanation: ZDT/Db2 was not connected to any Db2® system, and the user typed a new value in the SQLID field on the ZDT/Db2 main menu. It is not possible to change the current SQLID without first being connected to some Db2® system.

User response: Change the Db2® SQLID back to the default value, typically the user TSO logon ID. Now enter a valid Db2® SSID in the Db2® SSID field and press Enter to establish connection to the Db2® system. When the connection is successful, attempt to change the Db2® SQLID.

HFMDB059

** not defined in installation module.

Explanation: This message appears in the Db2® subsystem selection display against any Db2® system that has no HFM2SSDM macro entry in the HFM2POPT.

User response: No action required. This is an informational message. Refer to *Z Data Tools Customization Guide* for detailed information about how to code entries in the HFM2POPT module.

HFMDB060

Unable to load Db2® Interface module*name*. Abend Code *abend code-reason code*. Check STEPLIB/LINKLIST for Db2® load libraries.

Explanation: A Z Data Tools/Db2 batch job attempted to load the Db2® interface module *name*. The LOAD abended with Abend code *abend code*, reason *reason code*.

User response: Check the STEPLIB or JOBLIB statement in the batch job. Ensure that the appropriate Db2® load library is included. If the batch job was generated using ZDT/Db2 online, the Db2® libraries are automatically included, as specified in the HFM2SSDM macro entry in the HFM2POPT for the appropriate Db2® system. This will not occur if there is no HFM2SSDM macro entry for the Db2® system, or if the Db2® libraries are not included in the HFM2SSDM macro.

HFMDB061

Catalog table *table name* is missing.

Explanation: A Z Data Tools/Db2 batch job attempted to access Db2® catalog table *table name.*. The access failed with SQLCODE-204 (object not found).

User response: If views of the Db2® catalog are in use (CATOWNER=xxxx in the HFM2POPI macro, HFM2POPT module, xxxx is not SYSIBM) then ensure that the appropriate view xxxx.*table name* has been created.

HFMDB062

No select access to catalog table *table name*.

Explanation: A Z Data Tools/Db2 batch job attempted to access Db2® catalog table *table name*. The access failed with SQLCODE-551 (insufficient authority).

User response: Ensure that the Db2® authid associated with the batch job has SELECT access to table *table name*.

HFMDB063

Cannot access *column* in catalog table *table name*.

Explanation: A Z Data Tools/Db2 batch job attempted to access column *column* in Db2® catalog table *table name*. The access failed with SQLCODE-206 error. The column *column* is not defined in table *table name*.

User response: If views of the Db2® catalog are in use (CATOWNER=xxxx in the HFM2POPI macro, HFM2POPT module, xxxx is not SYSIBM) then ensure that the appropriate view xxxx.*table name* has been created. Further, ensure that column *column* is included in view xxxx.*table name*.

HFMDB064

Db2® Abend Code *abend code-reason code* accessing *name*. Notify your systems programmer.

Explanation: A Db2® Abend occurred while running a Z Data Tools/Db2 batch job. The Db2® Abend code is *abend code*, the reason code is *reason code* and the Db2® module that was accessed is *name*. The batch job ends and any uncommitted operations are backed out.

User response: Look up the Db2® Abend code and reason code in the DB2 for z/OS Messages and Codes. In most cases this type of error indicates an error with the batch job (for example, the wrong Db2® libraries being specified). Check the specification of the Db2® libraries in the HFM2SSDM macro entry for the appropriate Db2® system (HFM2POPT module). See the *Z Data Tools Customization Guide* online function, use ZDT/Db2 to regenerate the batch JCL and retry the operation.

HFMDB065

GEN command failed (short). The ZDT/Db2 GEN command failed - RC=*return code*. Please see your Systems Support staff for assistance. (long).

Explanation: The SQL Reverse Engineering function (GEN command) failed with reason *return code*.

User response: Possible failures include space related abends (x37) for the data sets used by the GEN code. If this is the case there will be an associated abend issued. Respond to the Abend by deleting and recreating the data set with more space, and retry the operation.

HFMDB066

No SSID parameter was passed to HFM2INEX. The last connected Db2® subsystem, if any, is used. Check that the connected Db2® subsystem is, in fact, what was intended.

Explanation: The REXX EXEC HFM2INEX can be used to invoke ZDT/Db2 by an external call. The REXX EXEC requires various parameters, including the Db2® SSID of the Db2® system that ZDT/Db2 should connect to. This was not provided.

User response: Check the method used to invoke the HFM2INEX exec and ensure that a Db2® SSID is provided.

HFMDB067

Invalid option code (short). The option code passed to HFM2INEX is either invalid or not supported. The option code has been discarded. (long).

Explanation: The REXX exec HFM2INEX can be used to invoke ZDT/Db2 by an external call. The exec requires various parameters, including a function code that indicates which ZDT/Db2 function to run. The function code provided was invalid.

User response: The function codes that may be passed to *Z Data Tools Customization Guide*. Ensure that a supported function code is specified.

HFMDB068

Command not active (short). The command to change the Db2® subsystem is not available when ZDT/Db2 is invoked remotely. (long).

Explanation: ZDT/Db2 was invoked from an external application. In this situation it is not possible to change the currently connected Db2® SSID.

User response: No action required. Informational message.

HFMDB069

Command not available (short). Db2® commands cannot be issued when ZDT/Db2 has been started remotely. (long).

Explanation: ZDT/Db2 was invoked from an external application. In this situation it is not possible to issue Db2® commands.

User response: No action required. Informational message.

HFMDB070

Possible causes - Missing/invalid HFM2POPT module, eg invalid plan name.

Explanation: A Z Data Tools/Db2 batch job encountered an error attempting to use the CAF attachment.

User response: The usual causes of this error are:

1. ZDT/Db2 is not installed on the target Db2® system.
2. The HFM2POPT module in use specifies an invalid plan name, perhaps the specified plan does not exist, or the plan exists but is invalid.

HFMDB071

Current SQLID (short). The current SQLID is the value specified. (long).

Explanation: The SQLID command was issued to change the current SQLID. However the specified SQLID is already the current SQLID.

User response: No action required. Informational message.

HFMDB072

SQLID value required (short). The SQLID command requires a value. Enter the new SQLID after the SQLID keyword, eg "SQLID FRANK". (long).

Explanation: The SQLID command was issued without an operand.

User response: Specify, for example, the desired SQLID after the SQLID command:

```
SQLID FRED
```

HFMDB073

SQLID cmd not available (short). The SQLID command is not available from this panel. (long).

Explanation: The SQLID command was issued from a panel where the command is not supported.

User response: Return to a higher level panel , for example a function entry panel, or menu panel, and reissue the command.

HFMDB074

UNICODE conversion error codepage *CCSID 1* to codepage *CCSID 2*.

Explanation: In a Z Data Tools/Db2 editor session of a Unicode encoded object, the conversion between CCSID *CCSID 1* and CCSID *CCSID 2* failed.

User response: One reason for this error is that the z/OS® Unicode translation service for the specified CCSID pair has not been set up. Contact your System Programmer. If this does not solve the problem, contact HCL Support.

HFMDB075

SQLID = *SQLID* (short). The current SQLID has been changed to*SQLID*. (long).

Explanation: A command to change the current SQLID was issued. The command succeeded, the current SQLID is now *SQLID*.

User response: No action required. Informational message.

HFMDB076

*** Rollback occurred. Transactions since start of session or last commit have been backed out. ***

Explanation: A Db2® rollback was issued. This message appears in audit log reports.

User response: No action required. Informational message.

HFMDB077

A Db2® commit has been issued.

Explanation: A Db2® commit was issued. This message appears in audit log reports.

User response: No action required. Informational message.

HFMDB078

UNICODE conversion error. UNICODE conversion error from codepage *CCSID 1* to codepage *CCSID 2*. Return code *return code*, reason code *reason code* from CUNLCNV routine.

Explanation: In a Z Data Tools/Db2 editor session of a Unicode-encoded object, the conversion between CCSID *CCSID 1* and CCSID *CCSID 2* failed. The CUNLCNV system service, used to perform the conversion, issued return code *return code* and reason code *reason code*.

User response: One reason for this error is that the z/OS® Unicode translation service for the specified CCSID pair has not been set up. Contact your System Programmer. If this does not resolve the problem, contact HCL Support.

HFMDB079

Change SSID first (short). An attempt to navigate to a Z Data Tools/Db2 function was made while simultaneously changing the Db2® SSID. This is not supported. First change the Db2® SSID and then navigate to the required function. (long).

Explanation: Processing on the ZDT/Db2 main menu panel includes the capability to change the current SQLID by typing a new value in the SQLID field, followed by the user pressing enter to validate the change. The ISPF short cut prefix ('=') can be used to navigate to some other ZDT/Db2 function. However, the use of this short cut bypasses the normal panel processing, and is therefore not supported.

User response: The use of '=' is not supported within *Z Data Tools Customization Guide* for Db2® data. In situations where a change in SQLID is required to execute some ZDT/Db2 function, first change the SQLID, then navigate to the required function.

HFMDB091

Invalid selection (short). Enter 'S' or '/' to select the required object type. (long).

Explanation: An invalid character was entered against an entry in an object selection list. The only characters that may be used to select the entry are 'S' and '/'.

User response: Use one of the characters indicated to select the object.

HFMDB092

Invalid selection (short). Enter 'S' or '/' to select the required line command. (long).

Explanation: An invalid character was entered against an entry in an object selection list. The only characters that may be used to select the entry are 'S' and '/'.

User response: Use one of the characters indicated to select the object.

HFMDB093

Invalid selection (short). Valid selections are 'S', '/', 'A', 'D', 'U', 'UO' and 'UW'. (long).

Explanation: This command is issued by the Basic SQL Prototyping function. An invalid command was entered in the 'S' (Selection) field. The only valid characters that may be used are as shown in the long message.

User response: Use one of the character or characters shown in the long message to select the column.

HFMDB094

OBJIN|OBJOUT Db2® object not found.

Explanation: This message is issued by ZDT/Db2 functions that are executed in batch, when a Db2® object name does not exist. OBJIN appears when an input Db2® object name does not exist, for example print, copy and export. OBJOUT appears when an output Db2® object name does not exist, for example copy, import and data create.

User response: Correct the Db2® object name, re-submit the batch job.

HFMDB095

OBJIN|OBJOUT Db2® object name error. The name specified returned multiple Db2® objects.

Explanation: This message is issued by ZDT/Db2 functions that are executed in batch, when the specified Db2® object name returns multiple Db2® objects. OBJIN appears when the location appears in an input Db2® object name, for example print, copy and export. OBJOUT appears when the location appears in an output Db2® object name, for example copy, import and data create.

User response: Correct the Db2® object name to specify a less generic Db2® object name, re-submit the batch job.

HFMDB096

OBJIN|OBJOUT Db2® location not found.

Explanation: This message is issued by ZDT/Db2 functions that are executed in batch, when a Db2® location, specified as part of a Db2® object name, does not exist. OBJIN appears when the location appears in an input Db2® object name, for example print, copy and export. OBJOUT appears when the location appears in an output Db2® object name, for example copy, import and data create.

User response: Determine the failing Db2® location value. Ensure that this Db2® location is accessible from the Db2® system specified in the PARM statement, re-submit the batch job.

HFMDB098

Db2 SSID; executing Db2® level *release*. Maximum supported level is *Maximum release*.

Explanation: ZDT/Db2 connected to *Db2 SSID*, which is at Db2® version or release *release*. The maximum supported Db2® version or release for the current version of ZDT/Db2 is *Maximum release*, which is less than *release*. The connection to this Db2® system is not supported.

User response: Contact HCL Support for information about which versions of ZDT/Db2 support Db2® systems at version or release *release*.

HFMDB099

Invalid Db2® SSID (short). The Db2® SSID is too long. Specify a 1-4 character value. (long).

Explanation: A command to change the currently connected Db2® system was issued, however the value entered for the desired Db2® system is too long. Valid Db2® SSIDs are 1-4 characters in length.

User response: Check the Db2® SSID that was specified; make corrections. Retry the operation.

HFMDB100

SQLCODE *SQLCODE*. *Statement type* (short). SQLCODE *SQLCODE* was encountered while issuing a *Statement type* statement.

Explanation: An error was encountered during execution of a*Statement type* SQL statement. The failing SQLCODE is *SQLCODE*.

User response: Look up the failing SQLCODE in the DB2 for z/OS Messages and Codes manual. If the message was issued when processing a user-specified SQL statement, make adjustments to the statement and retry the operation.

HFMDB101

SQL error message

Explanation: A Z Data Tools/Db2 batch function encountered an error during execution of an SQL statement. The formatted SQL error messages is shown in *SQL error message*.

User response: Look up the failing SQLCODE in the Db2® Messages and Codes manual. If the message was issued when processing a user-specified SQL statement, make adjustments to the statement and retry the operation.

HFMDB107

Remote server not found (short). ZDT/Db2 encountered an SQL error *SQLCODE* attempting to access the Db2® system at *Db2 remote server*. This might indicate that the remote Db2® system is not available or does not exist. Check the definitions in SYSIBM.LOCATIONS. (long).

Explanation: ZDT/Db2 attempted to connect to a remote Db2® server name, This may have been entered in the location field of various ZDT/Db2 function entry panels. The connection attempt failed with SQLCODE *SQLCODE*. The connection is not established and no operations at the remote Db2® server are possible until the reason for the connection failure is resolved.

User response: Look up *SQLCODE* in the DB2 for z/OS Messages and Codes manual and follow the recommended course of action. This error can occur in the following situations (non-exhaustive list):

1. DRDA® definition problems.
2. The remote Db2® system does not exist, is not active, is not available.
3. Communication problems.

HFMDB108

Different Db2® release (short). The release level of the local Db2® system is different to the release level of the Db2® system which you have just accessed. The choice of Db2® object types has been refreshed to show those object types supported on the remote or local system. Press ENTER again to get the requested list. (long).

Explanation: This message is issued by the ZDT/Db2 Object list utility when the user changes the value in the location field on the function entry panel and presses ENTER. ZDT/Db2 attempts to connect to the new location value (or to reconnect to the local server, if the new value is blank). When the Db2® version or release for the previously connected Db2® system is different to the newly connected Db2® system, it is possible that the list of Db2® object types requires changes. The object list request is processed in two steps:

1. Updating the list of objects and showing this message.
2. Displaying the list of objects for the selected object type.

User response: Ensure that the requested object type is still valid at the new Db2® system. Press **Enter** again to see the information for the requested object types.

HFMDB109

No matching locations (short). The location name specification does not match any locations defined to the local Db2® server. (long).

Explanation: A Db2® server name, including wild cards, was entered in the Location field on the ZDT/Db2 Object List utility function entry panel. There are no Db2® server names that match the wild card specification.

User response: Check the location specification and retry the operation. Use ***** in the location field to list all possible locations.

HFMDB110

Internal error (ABEND) *Abend code*. Function terminated.

Explanation: ZDT/Db2 detected a serious internal error and issued an internal ABEND to prevent further processing. The internal abend code is *Abend Code*

User response: This is an internal error. The list of Z Data Tools internal abend codes is not documented for external use. Record details of the circumstances that lead to the Abend, including the *Abend code*, and contact support.

HFMDB118

Function active (short). The command to change the Db2® subsystem is not available while a Z Data Tools/Db2 function is in progress.

Explanation: A command to change the currently connected Db2® SSID (for example SSID XXXX) was issued on a panel where the command is not available. The reason for this is that a Z Data Tools/Db2 function is active.

User response: Exit the current ZDT/Db2 function by returning to a function entry panel, or menu panel. Reissue the command to change the Db2® system.

HFMDB119

Db2 authid not authorized (short). You are not authorized to use the *Db2 authid* on Db2® Subsystem *Db2 SSID*. (long).

Explanation: An attempt was made to change the current SQLID, using the SQLID command. The user is not authorized to use Db2® authid *Db2 authid*. The attempt to change the current SQLID fails, the existing value for SQLID is retained.

User response: The rules governing the use of Db2® authids can be complex, depending on whether an external security server is used to control Db2® authorizations and whether a Db2® authorization exit is in use. In most cases a user needs SYSADM authority or similar to use other SQLIDs. Contact either your Db2® System Administrator, or the site Security Administrator for assistance.

HFMDB120

Inactivity time-out (Short). The editor session ended because of inactivity. Any uncommitted changes have been rolled back. Contact your System's Administrator for more information.

Explanation: There is a site-imposed limit on how long an ZDT/Db2 editor session can be inactive before it is canceled. This limit has been exceeded. The current editor session has been canceled, any uncommitted changed have been rolled back.

User response: None.

HFMDB145

Recursive alias (short). The object name entered is an alias that refers to an alias defined at the current server, which is not permitted. Specify an alias name that refers to an object, that is, not an alias. (long).

Explanation: A Db2® object name was entered in the Owner or Name fields on a Z Data Tools/Db2 function entry panel. The values entered refer to an ALIAS defined at the local Db2® server, however this ALIAS refers to another ALIAS. Recursive ALIASes are not supported.

User response: An ALIAS that refers to a Db2® object (Table/View/Synonym) is supported. Resolve the ALIAS to the point where the name entered refers to an actual object and specify that value. Alternatively, specify the target object name directly.

HFMDB147

Data sampling requires that ZDT/Db2 use a Db2 scrollable cursor, however, the use of Db2 scrollable cursors was disabled by the product installer. The operation cannot be performed. (long)

Explanation: An attempt was made to use data sampling, however the use of Db2 scrollable cursors has been disabled by the product installer. Data sampling requires the use of Db2 scrollable cursors so the requested operation cannot be performed.

User response: None.

HFMDB149

Count of unknown

Explanation: This message is issued by the ZDT/Db2 editor, when operating in Large mode. *Count* shows the row number of the first row currently displayed (in TABL display mode), or the currently displayed row (in SNGL display mode). Large editor mode is explained at length in ZDT/Db2 editor sessions on page 123. Large editor mode is selected when the user enters 0 (also *, ALL) in the "Row count" field of a Z Data Tools/Db2 function that uses the editor to display the result table.

User response: No action required. Informational message. In Large editor mode the number of rows in the result table is not available until the user scrolls to the end of the Db2® object. After the user scrolls to the end of the Db2® object, 'unknown' is replaced with the number of rows in the Db2® object/result table.

HFMDB150

Table locking disabled (short). Table locking was requested but failed, as can occur with eg catalog tables. (long).

Explanation: This message is issued by the ZDT/Db2 editor when the data is first displayed. The user specified a table locking option (not recommended) on the 7th Editor Options panel. When ZDT/Db2 attempted to lock the table, the operation failed.

User response: No action required. Informational message. This message can be issued when a Db2® catalog table is selected for edit.

HFMDB153

Commit issued (short). All pending changes to the database have been successfully committed. (long).

Explanation: This message is issued by the ZDT/Db2 editor (only for edit) when an explicit Db2® COMMIT has been issued. The Db2® COMMIT finalizes any pending changes made in the current editor session.

User response: No action required. Informational message.

HFMDB154

Commit issued (Errors) (short). Successful changes committed, unsuccessful changes remain pending. (long).

Explanation: This message is issued by the ZDT/Db2 editor (only for edit) when an explicit Db2® COMMIT has been issued. The Db2® COMMIT finalizes any pending changes made in the current editor session. The ZDT/Db2 editor processes changes data in an editor session on a row-by-row basis. It is therefore possible for some, but not all, changes to have been committed.

User response: No action required. Informational message. Use the 'e' prefix command to identify rows that are in error. If in doubt as to which changes have been made, exit the current editor session and restart.

HFMDB155

Key information on (short). Key and index information will be displayed in single format (long).

Explanation: The user entered the INDEXINF primary command in a Z Data Tools/Db2 editor session, and the prior state of the toggle was off. When the data is displayed in SNGL mode, an additional column appears on the left of the display, showing the columns that are part of keys (indexes) defined against the table.

User response: No action required. Informational message.

HFMDB156

Key information off (short). Key and index information will not be displayed in single format (long).

Explanation: The user entered the INDEXINF primary command in a Z Data Tools/Db2 editor session, and the prior state of the toggle was on. See also message HFMDB155 on page 1095.

User response: No action required. Informational message.

HFMDB157

Rollback issued (short). A Db2® rollback has been issued. Any uncommitted changes are lost. (long).

Explanation: This message is issued by the ZDT/Db2 editor when the user presses the PF key assigned to CANCEL, or types CANCEL, within the editor session. The current ZDT/Db2 editor session ends and an explicit Db2® ROLLBACK is issued. Any changes made after the last Db2® COMMIT are lost.

User response: No action required. Informational message.

HFMDB158

Auxiliary Table (short). Edit or browse of an auxiliary table is not possible. (long).

Explanation: An attempt was made to edit an auxiliary table. This action is not supported.

User response: No action required. Informational message. Only base Db2® objects can be edited.

HFMDB159

count rows fetched.

Explanation: This message is issued by the ZDT/Db2 editor when the data is first displayed. It shows the number of rows that have been fetched from Db2® and loaded into the current editor session. It indicates that the user specified a non zero "row count" value, that the specified number of rows has been loaded into the editor, and that the end of the Db2® object or result table was not reached.

User response: No action required. Informational message. Not all the rows in the Db2® table have been loaded into the ZDT/Db2 editor. Consequently, any edit operations apply only to those rows loaded.

HFMDB160

Insufficient storage (short). There was insufficient virtual storage to edit/browse this object. Use the Rows to Retrieve option or specify a row selection criteria to limit the amount of data retrieved from Db2®. (long).

Explanation: This message is issued by the ZDT/Db2 editor when there is insufficient memory available to load all the rows of a Db2® object or result table into memory. The editor operation cannot proceed.

User response: Review Restricting the data that ZDT/Db2 loads from Db2 on page 138, covering memory usage by the ZDT/Db2 editor. There are various options for resolving this issue, including (non-exhaustive list):

1. Increase the TSO Region Size.
2. Use the ZDT/Db2 editor in "Large" mode, to minimize memory usage.
3. Specify a lower "row count"value.
4. Use record selection criteria, WHERE clause, to reduce the number of rows that might be loaded.

HFMDB161

Warning: USA time format (short). The Db2® object being edited contains at least 1 column with the TIME data type and the system options are set to display these columns in USA format (for example, HH:MM AM|PM You may not be able to delete or update rows in this Db2® object because of the loss of information associated with the use of this format. See the Systems Options tutorial for more information. (long).

Explanation: This message is issued by the ZDT/Db2 editor when processing a Db2® object, or result, table that has at least one column with the TIME data type. In addition, the ZDT/Db2 system options are set to display TIME columns in USA format. This format truncates the seconds part of any TIME value, making it impossible to distinguish between, for example, 09:13:52 and 09:13:00. For this reason it may be impossible to update or change data for this table.

User response: If edit operations are required for this table, consider changing the ZDT/Db2 system options to display TIME columns in either ISO or JIS formats, both of which show the full seconds value. For more information about the TIME display formats supported by Db2®, see the DB2 for z/OS SQL Reference.

HFMDB163

Copy Pending (short). Updates are not allowed to tablespace *table space name* in *database name* because the tablespace is marked as COPY PENDING. In order to edit the table you should cancel the current edit session, then take action to remove the copy pending status. One way to do this is to run the Db2® COPY utility. See the Db2® Administration Guide for a full explanation. (long).

Explanation: This message is issued by the ZDT/Db2 editor when an attempt is made to edit a Db2® object that is marked as COPY PENDING. The Db2® object being edited is stored in table space *table space name* in *database name*. It is not possible

to change Db2® tables while they are in COPY PENDING status. COPY PENDING status can be set for a Db2® object after, for example, an initial data load, using the Db2® utilities, when certain utility options are specified.

User response: The recommended way to remove the COPY PENDING status is to run the Db2® COPY utility against the Db2® object. Consult your Db2® System Administrator for the best way to proceed.

HFMDB164

Syntax error (short). The WHERE clause contains a syntax error. (long).

Explanation: This message is issued by the template editor, or Column Selection, or Edit panel, when a syntax error is detected in the user-specified WHERE clause.

User response: Examine the WHERE clause for any possible errors. You can copy the complete SQL to either the clipboard, or a temporary data set, and then use an SQL processor such as ZDT/Db2 options 4.3, 4.4, or SPUFI to execute the statement. Any SQL errors for the statement will also be reported.

HFMDB165

Invalid column name (short). The WHERE clause contains an invalid column name, *column name*. (long).

Explanation: This message is issued by the template editor, or Column Selection/Edit panel, when the WHERE clause specifies an invalid column name. The reason is that the column name does not exist in the Db2® object or result table being processed.

User response: Check the WHERE clause and locate any occurrences of *column name*. Ensure that this column name is one of the columns in the Db2® object, or result table. Retry the operation.

HFMDB166

Open quoted string (short). The WHERE clause contains an open quoted string. (long).

Explanation: This message is issued by the template editor, or Column Selection/Edit panel, when the WHERE clause contains an open quoted string. Examples of open quoted strings are:

1. WHERE EMPNO = '
2. WHERE EMPNO ="

User response: Check the WHERE clause, and count the number of apostrophes and double quotation marks. An odd number for either indicates a problem. Make corrections and retry the operation. The rules for quoted strings can be found in the DB2 for z/OS SQL Reference.

HFMDB167

SQLCODE *SQLCODE* (short). Execution of the SQL statement will result in SQLCODE *SQLCODE*. (long).

Explanation: This message is issued by the template editor, or Column Selection and Edit panel, during validation of a user specified WHERE clause. The validation process used by ZDT/Db2 is to prepare a SELECT statement against the Db2® object being processed after appending any user specified WHERE clause. The prepare can fail for many reasons, for example syntax errors and incorrect column names. The prepare step may also fail for other reasons, in which case the failing SQLCODE is returned to the user for action.

User response: Look up the SQLCODE in the DB2 for z/OS Messages and Codes manual and respond as directed.

HFMDB170

Dependent rows (short). This related edit session shows only those rows that would be affected by the change to parent table *Parent table name*. The foreign key columns are marked with a K in the scale line, and are shown on the left of the display in table display format. In general, the foreign key value must be changed, or the row deleted, for the operation on the parent table to proceed.

Explanation: This message is displayed when the data for a new related edit session is first displayed, and the new table is a dependent table.

User response: The reason for editing a dependent table is to resolve foreign key values that are preventing the deletion of the corresponding primary key in the parent table. In some situations the dependent edit session will show all rows, not just those with the foreign key values that match the primary key value in the parent table.

HFMDB171

Parent table (short). In this related edit session the primary key columns are marked with a P in the scale line and are shown on the left of the display (table display format). You can use the IP prefix command to insert a new row with the primary key set to the foreign key value entered in the edit session of dependent table *dependent table*. (long).

Explanation: This message is displayed when the data for a new related edit session is first displayed, and the new table is a parent table.

User response: The reason for editing the parent table is to add a new primary key, so that a row in a dependent table with a new foreign key, that matches the primary key, can be added.

HFMDB173

All rows (short). This related edit session shows all rows of the table. Parent table *parent table name* is subject to a self referencing constraint such that deleting or changing a primary or parent key value may result in the recursive deletion of other rows in the parent table. ZDT/Db2 cannot determine which rows in table *dependent table name* will be affected by such an operation. (long).

Explanation: This message is displayed when the data for a new related edit session is first displayed, and the new table is a dependent table.

User response: The reason for editing a dependent table is to resolve foreign key values that are preventing the deletion of the corresponding primary key in the parent table.

HFMDB174

Commit issued (short). A Db2® commit has been issued. (long).

Explanation: This message is issued at the end of a Z Data Tools/Db2 editor session. It indicates that a Db2® COMMIT has been issued to finalize any pending changes to the Db2® object being processed.

User response: No action required. Informational message.

HFMDB175

Invalid value (short). Valid values are TABL and SNGL. There are no unformatted edit modes in ZDT/Db2. (long).

Explanation: The user typed an invalid value in the Format field in a Z Data Tools/Db2 editor session.

User response: Check the value that was entered. There are two display modes when using the ZDT/Db2 editor:

1. TABL (Can be abbreviated to T, TA, TAB). In this mode each horizontal line on the screen shows one row of data from the Db2® object or result table being processed.
2. SNGL (Can be abbreviated to S, SN, SNG). In this mode the data for a single row of the Db2® object or result table being processed is displayed. The data for each column is shown, with the column name on the left and the data for that column on the right.

HFMDB176

Object/Alias not found (short). The specified object or alias was not found in the catalog, or the target of an alias does not exist.

Explanation: The user entered the name of a Db2® object in the "Owner" and "Name" fields on a Z Data Tools/Db2 function entry panel. ZDT/Db2 attempted to resolve the name and the attempt failed. The reason this message is issued is described in the following example:Alias *ALIAS* is defined, this refers to *OBJECT*. However, *OBJECT* does not exist. The user enters ALIAS in the "Owner" and "Name" fields on a Z Data Tools/Db2 function entry panel. ZDT/Db2 resolves the ALIAS name to *OBJECT*. However, the latter does not exist, resulting in this error message.

User response: Determine why the Db2® Object Name does not exist.

HFMDB177

Warning: USA time format (short). The Db2® object being exported contains at least 1 column with the TIME data type and the system options are set to externalize these columns in USA format (HH:MM *AM|PM)*. The exported data will not contain seconds information because of the loss of information associated with the use of this format. See the Systems Options tutorial for more information. (long).

Explanation: This message is issued by either the ZDT/Db2 export, or ZDT/Db2 UNLOAD utilities when:

1. The Db2® object being processing has a TIME data type column.
2. The "External Format for TIME Data Type Columns" system option is set to"Db2® DSNHDECP setting".
3. The DSNHDECP setting for TIME columns is 'USA'.

User response: This is an informational message. As noted in the long message text, in this situation the exported data for the TIME columns will lose any seconds value. If this is intended or desired, then no action is required. When no loss of information is required, consider changing the "External Format for TIME Data Type Columns" value to either 'ISO' or 'JIS', both of which include the seconds information.

HFMDB178

Warning: USA time format (short). The Db2® object being copied contains at least 1 column with the TIME datatype and the system options are set to externalize these columns in USA format (HH:MM *AMPM*). The data copied to the target table will not contain second's information because of the loss of information associated with the use of this format. See the Systems Options tutorial for more information. (long).

Explanation: This message is issued by the ZDT/Db2 copy utility when:

1. The Db2® object being processing has a TIME data type column.
2. The"External Format for TIME Data Type Columns" system option is set to "Db2® DSNHDECP setting".
3. The DSNHDECP setting for TIME columns is 'USA'.

User response: This is an informational message. As noted in the long message text, in this situation the copied data for the TIME columns will lose any seconds value. If this is intended or desired then no action is required. When no loss of

information is required, consider changing the "External Format for TIME Data Type Columns" value to either 'ISO' or 'JIS', both of which include the seconds information.

HFMDB179

Cancel recommended (short). Db2® has rolled back the current unit of work (SQLCODE -911) because of a deadlock with another task. ALL UNCOMMITTED CHANGES made during the current edit session have been lost. Consequently, the data displayed in the current edit session may not reflect the current state of the Db2® table. It is recommended that you cancel the current edit session and restart with a fresh copy of the data. (long).

Explanation: This message is issued by the ZDT/Db2 editor when it receives an SQLCODE-911 error attempting to change data for a row that was modified by the user. Db2® detected a deadlock with another task and resolved the situation by rolling back the changes for the ZDT/Db2 user. This means that any changes made since the last Db2® COMMIT point have been lost. In this situation there is no guarantee that the data displayed in the ZDT/Db2 editor still reflects the data stored in Db2®. It is best to cancel the current editor session and restart.

User response: Cancel the current edit session and restart. Make a change to a single row and attempt to save the change. If the SQLCODE-911 reoccurs, contact your Db2® System Administrator for assistance in determining why a deadlock is occurring.

HFMDB180

Related edit failed (short). Initialization of the related edit session failed. The reason is that the product has not been installed correctly. If LIBDEFs are used to allocate the Z Data Tools Db2 component libraries, the related edit session exec, HFM2RESS, should be installed in the same library as the exec used to allocate the libraries for, and start, the first Z Data Tools Db2 component session. Refer to the Z Data Tools Customization Guide for further information. (long).

Explanation: The initialization process for the related edit session failed. Each related edit session is effectively a new invocation of ZDT/Db2. Once started the user can use ISPF commands to switch between the different logical sessions. During the invocation process a REXX EXEC, HFM2RESS, is called. However, this REXX EXEC may not be accessible when LIBDEFs are used to define the ZDT/Db2 libraries.

User response: Contact the person who installed ZDT/Db2.

HFMDB181

Browse substituted (short). The object being edited is a read-only view. In Db2® parlance a read-only view is any view that has at least one column marked as not updateable in the catalog. For this view every column is marked as not updateable. Consequently no edit operations of any kind are possible, and browse has been substituted for edit. (long).

Explanation: This message is issued by the ZDT/Db2 editor when processing a view that cannot be updated. While the user selected the edit (as opposed to view/browse) to process the object, no edit operations are possible. To avoid any possible confusion the edit session has been converted to browse.

User response: No action required. Informational message.

HFMDB182

Table locking failed (short). Table locking was requested but failed. The object being edited is a view, and views cannot be explicitly locked. (long).

Explanation: This message is issued by the ZDT/Db2 editor when the data for a view is first displayed. The user has set a table locking option in the editor options. This is not recommended. Only tables can be explicitly locked, therefore the attempt to lock the view failed.

User response: No action required. Informational message. Table locking should be avoided because of the potential for deadlocks.

HFMDB183

Read-only view (short). The object being edited is a read-only view. In Db2® parlance a read-only view is any view that has at least one column marked as not updateable in the catalog. For this view at least one column is marked as updateable. Those columns that are not updateable are protected. Some edit operations may fail. (long).

Explanation: The object being processed by the ZDT/Db2 editor is a read-only view, although it does have one or more columns marked as updateable.

User response: This is an informational message. It is possible to successfully edit some read-only views, even though some columns of the view cannot be updated. However, attempting to edit a read-only view is not advisable. Edit the underlying tables directly.

HFMDB184

Non-updateable columns (short). The table being edited contains at least one column marked as not updateable in the Db2® catalog. This can occur when editing Db2® catalog tables. The not updateable columns are protected. Some edit operations may fail. (long).

Explanation: The object being edited contains one or more columns marked as not updateable in the Db2® catalog SYSIBM.SYSCOLUMNS.UPDATES = 'Y'. These columns will appear in the editor session as protected fields. One reason for this message is when edit is used to display the data from a Db2® catalog table.

User response: This is an informational message. You will not be able to change the data for those columns marked as not updateable. It may also not be possible to delete or insert rows.

HFMDB185

Recursive alias (short). The object name entered is an alias that refers to an alias at a remote location, which is not permitted. Specify the alias name at the remote server directly. (long).

Explanation: The user entered an alias name in the "Owner" and "Name" fields on a Z Data Tools/Db2 function entry panel. ZDT/Db2 resolved the alias name to another alias name defined at a remote location. This is not supported. After resolution, any alias name entered must refer to an actual Db2® object defined either locally, or remotely.

User response: Enter the target Db2® object name directly, or ensure that any alias used refers directly to a Db2® object.

HFMDB186

Not available (View) (short). The RE command is not available when editing a Db2® view. Edit the Db2® table directly to access related table edit processing. (long).

Explanation: The user entered the RE (Related edit) command in a Z Data Tools/Db2 editor session. The object being edited is a view; related edit functionality is only available for tables. The operation cannot be performed.

User response: If related edit processing is required, ensure that all objects referred to are tables.

HFMDB187

Command not available (short). Commands such as INSERT, REPEAT and COPY are not available in this edit session. The data for the Db2® object is stored in UNICODE format, but there is no conversion routine for graphic data type columns. (long).

Explanation: This message is issued by the ZDT/Db2 editor when processing a Db2® object with graphic data type columns that is encoded in Unicode. In this situation it is possible for data corruption to occur when processing data for these columns and initializing the field contents. To avoid this situation, the INSERT, REPEAT, and COPY commands are disabled.

User response: No action required. Informational message.

HFMDB188

The Db2® object being edited is stored in a UNICODE-encoded table space, and the object contains MBCS (mixed byte character set) data (GRAPHIC data type columns). The current application encoding scheme is not a known MBCS CCSID. Changing the GRAPHIC data type columns is not supported. (long).

Explanation: This message is issued by the ZDT/Db2 editor when processing a Db2® object with graphic data type columns that is encoded in Unicode. In this situation it is possible for data corruption when processing data for these columns. To avoid this possibility, changes to graphic data type columns are not supported.

User response: No action required. Informational message.

HFMDB189

Incompatible options (short). Read-only access can only be selected with concurrency options that do not specify that locks will be kept, that is all options except 4 "Read stability keep locks" and 6 "Repeatable read keep locks". (long).

Explanation: There is a conflict between options selected on the 7th Editor options panel. The "read-only" option implies that no Db2® locks are required when processing the SQL SELECT statement. However, the Concurrency options "Read stability keep locks" and "Repeatable read keep locks" imply that Db2® should take and keep locks while processing the SQL SELECT statement.

User response: In all but exceptional cases the Concurrency option should be set to "Uncommitted read" or "No with clause". If a concurrency option that requires locks is required, do not specify the "Read-only access" option.

HFMDB190

Statement overflow (short). The SQL statement needed to update the table is too long. (long).

Explanation: An internal statement buffer overflow was detected while building an SQL statement to update a table. The operation cannot be performed. This condition should not occur during processing of typical Db2® objects. It might occur when processing Db2® objects with very many column names (the maximum is 750), and with many "long" column names. The maximum length for a column name is 30 characters.

User response: Collect information including the DDL used to create the Db2® object being processed. Contact HCL Support.

HFMDB191

Changes discarded (short). The invalid change to the Initial Display option has been discarded. (long).

Explanation: The user made an invalid change to the "Initial display format" editor option, and then pressed the PF key assigned to END, or typed END. The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDB192

Changes discarded (short). The invalid change to the Concurrency option has been discarded. Enter one of the listed values for the Concurrency option. (long).

Explanation: The user made an invalid change to the "Concurrency" editor option, and then pressed the PF key assigned to END, or typed END. The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDB193

Changes discarded (short). The invalid change to the Enter key usage option has been discarded. Enter one of the listed values for the Enter key usage option. (long).

Explanation: The user made an invalid change to the "Enter key usage" editor option, and then pressed the PF key assigned to END, or typed END. The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDB194

Use unique characters (short). The character used to indicate the "String delimiter (display)" must be different to the character used to indicate the "String delimiter (input)". Select a different character. (long).

Explanation: The user specified the same character for the "Input" and "Display" string delimiters on the 5th Editor options panel. While most characters can be used, the characters that are used must be distinct.

User response: Ensure that the characters specified for the "Input" and "Display" indicators and delimiters are unique.

HFMDB195

Use unique characters (short). The character used to indicate the;"Null column display indicator" must be different to the character used to indicate the "Null column input indicator" Select a different character. (long).

Explanation: The user specified the same character for the "Input" and "Display" Null column indicators on the 5th Editor options panel. While most characters can be used, the characters that are used must be distinct.

User response: Ensure that the characters specified for the "Input" and "Display" indicators and delimiters are unique.

HFMDB196

Use unique characters (short). The character used to indicate the "Null column input indicator" must be different to the character used to indicate the "String delimiter (input)". Select a different character. (long).

Explanation: The user specified the same character for the "Input" Null column indicator and "Input" String delimiter. While most characters can be used, the characters that are used must be distinct.

User response: Ensure that the characters specified for the "Input" and "Display" indicators and delimiters are unique.

HFMDB197

Use unique characters (short). The character used to indicate the "Null column display indicator" must be different to the character used to indicate the "String delimiter (input)". Select a different character (long).

Explanation: The user specified the same character for the "Display" Null column indicator and "Input" String delimiter. While most characters can be used, the characters that are used must be distinct.

User response: Ensure that the characters specified for the "Input" and "Display" indicators and delimiters are unique.

HFMDB198

Use unique characters (short). The character used to indicate the "Null column input indicator" must be different to the character used to indicate the "String delimiter (display)". Select a different character (long).

Explanation: The user specified the same character for the "Input" Null column indicator and "Display" String delimiter. While most characters can be used, the characters that are used must be distinct.

User response: Ensure that the characters specified for the "Input" and "Display" indicators and delimiters are unique.

HFMDB199

Use unique characters (short). The character used to indicate the "Null column display indicator" must be different to the character used to indicate the "String delimiter (display)". Select a different character. (long).

Explanation: The user specified the same character for the "Display" Null column indicator and "Display" String delimiter. While most characters can be used, the characters that are used must be distinct.

User response: Ensure that the characters specified for the "Input" and "Display" indicators and delimiters are unique.

HFMDB206

Not on remote object (short). The requested function can not be performed on an object at a remote location. (long).

Explanation: This message is issued by the ZDT/Db2 Object List utility in various circumstances when a requested command cannot be performed. The reason the command cannot be performed is that the object is at a remote location. Certain SQL commands, for example GRANT and REVOKE, may only be issued when connected to the local Db2® server.

User response: Connect to the remote Db2® server directly and reissue the command.

HFMDB208

Not on catalog object (short). The requested function can not be performed on a Db2® catalog object. (long).

Explanation: The LISTCAT line command was issued against a Db2® catalog object. For example, a table or index space. This command cannot be issued against Db2® catalog objects.

User response: No action required. Informational message.

HFMDB210

Invalid value (short). The valid values are 16, for an 8 byte number with 16 digit coefficient, or 34 for a 16 byte number with 34 digit coefficient. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions. For a column defined with the DECFLOAT data type, a value other than 16 or 34 was entered in the 'PR' column.

User response: Enter either 16 for an 8-byte number, or 34 for a 16-byte number.

HFMDB215

Unsupported data type (short). This Z Data Tools Db2® function does not support columns of the specified data type. Supported data types are *List of data types*. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions. When defining the data type for a column, an invalid, or unsupported, data type value was entered.

User response: Check that the data type entered is spelled correctly. The list of supported data types is shown in the long version of the message. Press the PF key assigned to HELP, PF1 by default, to access. Check that the data type is supported by the version of Db2® in use. For example, the DECFLOAT data type is not supported at Db2® version 8.

HFMDB216

Length error (short). The specified length, precision, or scale is not valid for the data type of the column. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions. The value entered for a length, precision or scale is invalid.

User response: Check the value entered. Consult the DB2 for z/OS SQL Referencemanual for detailed information on the permissible values for data type lengths, and the scale and precision values that may be specified.

HFMDB217

Scale > precision (short). The scale of a decimal must be less than or equal to the precision of the number. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions. The combination of scale and precision values entered is invalid.

User response: For Decimal numbers the precision can be 1 to 31, inclusive. The scale for a number can be 0 to p, where p is the precision of the number.

HFMDB220

Non updateable field (short). A field reference in a change command refers to a column that is not updateable. (long).

Explanation: In the ZDT/Db2 editor, a change command was issued that included a field reference, or references. The target column for the change is not updateable, meaning that the proposed change is not possible.

User response: Ensure that the field reference identifies a column that can be changed. Non updateable columns are shown as protected fields in the editor display.

HFMDB223

Start too large (short). The start position (*start position*) is larger than the number of rows (*number of rows*) in the result table.

Explanation: This message is issued by the ZDT/Db2 editor. The user entered a value *start position* in the "Start position" field on the Db2® Edit panel. When ZDT/Db2 attempted to skip to the nominated start position, it reached the end of the result table. The number of rows in the result table is *number or rows*.

User response: Informational message. Reduce the start position value to at least *number of rows* and retry the operation.

HFMDB224

Start out of range (short). The first sample row is at row *row number*, which is larger than the number of rows, *number of rows*, in the result table. Reduce the start position and/or initial skip count values.

Explanation: This message is issued by the ZDT/Db2 editor when clustered sampling is selected in the editor options. When ZDT/Db2 attempted to skip to the nominated start position, determined by the value in the "start position field", and the value in the "Initial skip count fields", it reached the end of table. The number of rows in the result table is *number of rows* (n).

User response: Informational message. Reduce either or both of the "start positions" and "Initial skip count" (i) values, so that s+i+1 <=n.

HFMDB240

Invalid collection (short). The collection must be different from the COPY collection. (long).

Explanation: When binding a copy of a package, the collection and COPY collection values must be different.

User response: Specify different values.

HFMDB242

DASD data set required (short). This command can only be used on a row where DSNAME refers to a DASD data set. (long).

Explanation: The 'B' (Browse) command was issued against an image copy entry displayed by the ZDT/Db2 Object List Utility. The browse command can only be entered against entries associated with an image copy data set stored on DASD (disk). The command cannot be processed.

User response: Issue the command against an image copy data set with a non blank device type. The device type must not be a tape type.

HFMDB245

No plan (short). The command is only valid when application plan is non-blank. (long).

Explanation: The DBRM command was entered against an entry in a PLAN table. The command is only valid when the entry is associated with an application plan; that is the APPLNAME column for the row is non blank. The command cannot be processed.

User response: Ensure that the APPLNAME column value is non blank.

HFMDB246

No collection (short). The command is only valid when collection is non-blank. (long).

Explanation: The PACKage command was entered against an entry in a PLAN table. The command is only valid when the entry is associated with a package; that is the COLLID column for the row is non blank. The command cannot be processed.

User response: Ensure that the COLLID column value is non blank.

HFMDB247

No DBRM (short). The command is only valid when DBRM is non-blank. (long).

Explanation: The PACKage command was entered against an entry in a PLAN table. The command is only valid when the entry is associated with a DBRM; that is the PROGNAME column for the row is non blank. The command cannot be processed.

User response: Ensure that the PROGNAME column value is non blank.

HFMDB250

Invalid SQL identifier (short). The value entered is not a valid SQL ordinary identifier. (long).

Explanation: This message is issued in various places in the ZDT/Db2 Object List Utility. A value was entered for a field where an ordinary SQL Identifier is required. An example is a Db2® authid used as an "Owner" or SQLID.

User response: An SQL ordinary identifier cannot include characters other than A-Z, 0-9, _ ,and the national characters, and must not begin with a digit. See the Db2® SQL Reference manual for a precise definition. Refer to the Appendix "Limits in Db2® for z/OS®", for a complete list of valid values for various types of Db2® items.

HFMDB270

Invalid buffer pool (short). Valid buffer pool names are: *List*. (long).

Explanation: A value entered for a buffer pool is invalid. *List* shows the valid values that can be entered.

User response: Compare the value entered against the values in *List*. Change the value entered to match one of the values in *List*. Retry the operation.

HFMDB271

Db2® version invalid (short). Table constraints are only available when connected to a Db2® Version 8 or later system. Current Db2® version is *Db2 VRM indicator*. (long).

Explanation: The "CON" command (Show table constraints) was issued against a list of tables in the ZDT/Db2 Object List utility. This command is not valid when connected to a Db2® version 7, or earlier system. The currently connected Db2® system is at *Db2 VRM indicator*, which is less than '0800'. The command cannot be executed.

User response: No action required. Informational message.

HFMDB272

Not valid for object (short). "*line command*" is not a valid command for this type of object. (long).

Explanation: This message is issued in various contexts by the ZDT/Db2 Object List utility. An example is issuing the VS (Show view SQL) against a entry in a list of Db2® tables/views/aliases, when the entry is not a view.

User response: Ensure that the command entered is appropriate for the type of object shown in the row.

HFMDB273

Not valid for function (short). "*Line command*" is only valid for external user-defined functions. (long).

Explanation: This message is issued in various contexts by the ZDT/Db2 Object List utility when processing a list of functions. The row against which the line command was entered is of the wrong function type. The command is not applicable to that type of function.

User response: The type of function is shown in the "Type of Routine" column. Enter the command against a row showing a function of the appropriate type.

HFMDB274

No upgrade done (short). The PLAN_TABLE was already at the correct level. (long).

Explanation: A command was issued to upgrade a PLAN_TABLE, however the table was already at the correct level. No action was taken.

User response: No action required. Informational message.

HFMDB275

Not valid for column (short). "*Line command*" is not a valid command for this type of column. *List of types*. (long).

Explanation: This message is issued in various contexts by the ZDT/Db2 Object List utility when processing a list of columns. The row against which the line command was entered is of the wrong type. For example, the command applies only to columns defined with a user-defined (DISTINCT) data type. The command cannot be executed.

User response: No action required. Informational message. *List of types* contains information about the types of columns that the command can be issued against.

HFMDB276

Not valid for column (short). *Line command* is not a valid command for 'identity column' type sequence objects. (long).

Explanation: The 'COM' (COMMENT) line command was issued against a row in a list of sequences displayed by the ZDT/Db2 Object List Utility. A comment cannot be specified for sequences that represent internal sequence objects used by Db2®. The command cannot be executed.

User response: No action required. Informational message.

HFMDB289

Data set required (short) This command can only be used on a row where DSNAME refers to a data set. (long).

Explanation: The user issued the 'LC' (LISTCAT) command against an entry in a list on the Recovery Information panel in the ZDT/Db2 object list utility that did not have an associated data set. The Recovery Information panel shows all the rows in the SYSIBM.SYSCOPY catalog table. Only rows that represent full and incremental image copies are suitable targets for the LISTCAT command.

User response: No action required. Informational message. Ensure that the selected row represents either a full or incremental copy. The command may still fail if, for example, the data set has been deleted.

HFMDB292

Invalid criteria (short). The criteria specified in the Additional Selection Criteria fields produced an invalid SQL SELECT statement WHERE clause. Db2® returned an SQL code of -199 to the SQL statement *"SQL STATEMENT"*. (long).

Explanation: This message is issued by the ZDT/Db2 object list utility when the user has entered values in the "Additional selection criteria" section of the panel. This section of the panel can be used to add a simple predicate to the SQL being generated. For example, when listing tables, show only tables that have an EDITPROC. This can be achieved by setting Column=EDPROC; Operator= ≠ Value = ''. The expression entered resulted in SQLCODE-199, and cannot be processed.

User response: Look up SQLCODE-199 in the DB2 Administration Tool for z/OS User's Guide and Reference, modify the predicate, and retry the operation.

HFMDB295

Invalid location (short). The location name is invalid or is not defined to the Db2® subsystem. (long).

Explanation: The user entered a value in a Location field on a Z Data Tools/Db2 function entry panel. The attempt to connect to this remote server failed with SQLCODE-950.

User response: Look up SQLCODE-950 in the Db2® SQL Messages and Codes manual. The most likely explanation is that the value entered does not appear in column SYSIBM.LOCATIONS.LOCATIONS at the local Db2® server. If the value is incorrect then amend it and retry the operation. If the value is correct contact the person responsible for maintaining DRDA® definitions for your Db2® systems.

HFMDB296

No plan table (short). Plan table *owner*.PLAN_TABLE does not exist. (long).

Explanation: This message is issued by the Explain Utilities functions when the user issues the 'L' (List plan table rows) command, but the plan table *owner*.PLAN_TABLE does not exist. The operation cannot be performed.

User response: The PLAN_TABLE is a user-defined table. The explain utilities allow an "owner" value to be specified. Check that this value is not preventing access to the user's PLAN_TABLE. All PLAN_TABLE entries can be listed using the ZDT/Db2 Object List utility. If the PLAN_TABLE does not exist, use the 'C' command to create it.

HFMDB298

Incomplete quoted string (short). You entered an incomplete quoted string. Db2® returned an SQL code of -010 to the SQL statement *SQL Statement*. (long).

Explanation: This message is issued when a generated SQL statement contains a syntax error, specifically an open quoted string. The SQL statement cannot be executed.

User response: Examine the SQL statement to determine the source of the open quoted string. Modify the values entered to avoid the problem, retry the operation.

HFMDB299

Not COPY FULL (short). The RT command is only valid against an image copy entry. (long).

Explanation: The user issued the 'RT' (Recover TOCOPY) command against an entry in a list on the Recovery Information panel in the ZDT/Db2 object list utility that did not have an associated data set. The Recovery Information panel shows all the rows in the SYSIBM.SYSCOPY catalog table. Only rows that represent full and incremental image copies are suitable targets for the RT command.

User response: Ensure that the row selected represents either a full or incremental image copy.

HFMDB300

No previous template has been used with this Db2® object. (long).

Explanation: This message is issued by a Z Data Tools/Db2 function entry panel when the user specifies "Previous" for the Template usage option. However, no template has been previously saved for the current Db2® object.

User response: You can generate and save a template for the current Db2® object using the "Generate/Replace" template usage option. Once a template has been successfully saved, use the "Previous" option to access the template, without needing to enter a template data set name or member.

HFMDB301

Invalid ZDT/Db2 template (short). The specified template does not represent a ZDT/Db2 template. Specify a valid ZDT/Db2 template. (long).

Explanation: This message is issued by a Z Data Tools/Db2 function entry panel when the user specifies a template data set name for a Db2® object, and that the specified template should be used. However, the specified template is not a valid ZDT/Db2 template. There are different types of Z Data Tools templates, depending on the source of the template. A base template cannot be used to describe a Db2® object.

User response: You can generate and save a template for the current Db2® object using the "Generate/Replace" template usage option. Once a template has been successfully saved, you can access it and possibly modify it using the "Edit template" option on the appropriate ZDT/Db2 function entry panel.

HFMDB302

ZDT/Db2 template loaded (short). Z Data Tools Db2 component template has been successfully loaded. (long).

Explanation: This message is issued by a Z Data Tools/Db2 function entry panel when the user specifies a template data set name for a Db2® object, and that the specified template should be used when processing the Db2® object. The specified template was successfully loaded, compared against the Db2® object and found to contain a valid description of the Db2® object.

User response: No action required. Information message.

HFMDB303

Template not loaded (short). Loading of ZDT/Db2 template canceled by user. Row selection criteria for the template must be revised before proceeding. (long).

Explanation: This message is issued by a Z Data Tools/Db2 function entry panel when the user specifies a template data set name for a Db2® object, and that the specified template should be used when processing the Db2® object. When ZDT/Db2 compared the template specified by the user with the template generated by ZDT/Db2, using the information in the Db2® catalog, differences were found. The "Template not matched" panel is typically shown before this message; the user pressed PF3 on the "Template not matched" panel. In most cases this error means that the template in the user-specified data set refers to a different Db2® object to that specified on the function entry panel. See also message HFMDB304.

User response: Resolve the reason for the difference between the data set template and the Db2® object, and retry the operation. Note that ZDT/Db2 always generates a template for any Db2® object and compares this with any user-specified template; with the former template considered to be correct. There are many reasons why the two templates may be inconsistent (for example, Different Db2® objects; Object dropped changed and recreated).

HFMDB304

Template mismatch (short). The selected template does not match the Db2® object. Press ENTER to replace selected template, or PF3/END to select another template or processing option. (long).

Explanation: This message is issued by a Z Data Tools/Db2 function entry panel when the user specifies a template data set name for a Db2® object, and when the user specifies the template should be used when processing the Db2® object. When ZDT/Db2 compared the template specified by the user with the template generated by ZDT/Db2, using the information in the Db2® catalog, differences were found. The "Template not matched" panel is typically shown at the same time as this message.

User response: There are two choices: 1. Press PF3 to cancel the current function. You may then attempt to resolve why the specified template data set does not match the Db2® object and retry the operation. 2. Press ENTER to replace the template in the data set with a correct version for the Db2® object. Any customization stored in the data set template will be lost. Note that ZDT/Db2 always generates a template for any Db2® object and compares this with any user-specified template; with the former template considered to be correct. There are many reasons why the two templates may be inconsistent (for example, Different Db2® objects; Object dropped changed and recreated).

HFMDB305

Template not updated (short). Loading of Z Data Tools Db2 component template canceled by user as the specified template does not match the Db2® object. Specify another template or processing option. (long).

Explanation: This message is issued by a Z Data Tools/Db2 function entry panel when the user specifies a template data set name for a Db2® object, and that the specified template should be used when processing the Db2® object. When ZDT/Db2 compared the template specified by the user with the template generated by ZDT/Db2, using the information in the Db2® catalog, differences were found. The user canceled the template merge process.

User response: No action required. Informational message.

HFMDB309

The object you are attempting to edit is a Materialized Query Table and it was not configured with the MAINTAINED BY USER option. As a result, the EDIT/VIEW session is substituted with a BROWSE session.

Explanation: This message is issued in a Z Data Tools/Db2 VIEW/EDIT session when the user attempts to edit a Materialized Query object that has not been configured with the **DEFERRED** refresh mode and with the **MAINTAINED BY USER** option. The user cannot update this data and the data is displayed in a Z Data Tools/Db2 BROWSE session instead.

User response: No action required. Informational message.

HFMDB310

No statement table (short). Plan table *Owner*.DSN_STATEMNT_TABLE does not exist. (long).

Explanation: This message is issued when an attempt is made to access *Owner*.DSN_STATEMNT_TABLE. The user-defined table does not exist, the operation cannot be performed.

User response: Create the table using the ZDT/Db2 Explain Utilities function. After populating the table using the EXPLAIN command, retry the operation.

HFMDB311

No function table (short). Plan table *Owner*.DSN_FUNCTION_TABLE does not exist. (long).

Explanation: This message is issued when an attempt is made to access *Owner*.DSN_FUNCTION_TABLE. The user-defined table does not exist, the operation cannot be performed.

User response: Create the table using the ZDT/Db2 Explain Utilities function. After populating the table using the EXPLAIN command, retry the operation.

HFMDB330

Invalid row number (short). Either a row number has not been specified, or a specified value is invalid (long).

Explanation: The user entered a command to process a LOB or XML column. The row number specified as part of the command is invalid. You can use cursor positioning to specify the row in which the column of interest is located - in which case a row number is not required. If cursor positioning is not used a row number is required. This message can be issued when an invalid value (not a number) is specified for the row number.

User response: Check the syntax for the command and re-issue it.

HFMDB331

Row *Row* not found (short). The specified row number *Row* does not exist in the Db2® object, or result table. (long).

Explanation: The user entered a command to process a LOB or XML column. The row number specified as part of the command is invalid because it does not exist in the current editor session. The command cannot be processed.

User response: Check the row number value and ensure that it is between 1 and the maximum number of rows in the current editor session.

HFMDB332

Not on XML/LOB column (short). The cursor is not positioned on an XML/LOB column. (long).

Explanation: The user entered a command to process a LOB or XML column. The cursor was positioned within the data area of the display and ZDT/Db2 attempted to use cursor positioning to identify the column and row of interest. The column where the cursor is located in not a LOB or XML column. The command cannot be processed.

User response: Check the cursor position and ensure that it is located on a column of the appropriate type for the command.

HFMDB333

J command invalid (short). The J command is only valid in a Z Data Tools/Db2 editor session of a LOB. (long).

Explanation: The user entered the J prefix command in a Z Data Tools edit or view session. This command is only available in a Z Data Tools/Db2 component editor session when processing a LOB, that is in a view or edit session initiated using the LOBVIEW or LOBEDIT command. The command cannot be processed.

User response: None - informational message.

HFMDB334

SP command invalid (short). The SP command is only valid in a Z Data Tools/Db2 editor session of a LOB. (long).

Explanation: The user entered the SP prefix command in a Z Data Tools edit or view session. This command is only available in a Z Data Tools/Db2 component editor session when processing a LOB, that is in a view or edit session initiated using the LOBVIEW or LOBEDIT command. The command cannot be processed.

User response: None - informational message.

HFMDB335

Large table mode is not available when the object being processed contains LOB or XML columns. Specify a row count greater than zero to process this Db2® object. (long).

Explanation:

This message is issued on the ZDT/Db2 editor function entry panel when:

1. The user enters the name of a Db2® object that contains a LOB or XML column.
2. The row count value is set to 0 (also * or ALL).

This combination is not supported. Large editor mode is not available when the Db2® object contains LOB or XML columns.

User response: Change the row count to specify a positive integer value.

HFMDB336

Non-updateable field (short). LOB (BLOB, CLOB, DBCLOB) and XML fields cannot be altered using the CHANGE command. Use LOBEDIT or XMLEDIT to display, and possibly change, the contents of LOB/XML columns.

Explanation: A change command was entered with a target column that is a LOB (CLOB, BLOB or DBCLOB) or XML column. This is not supported

User response: Use the LOBEDIT or XMLEDIT commands to edit the LOB/XML data.

HFMDB337

LOB/XML Columns (short). The result table for the SELECT statement contains either LOB or XML columns. Display of the result table is not supported. Use Browse, View or Edit of the Db2® object containing the LOB/XML columns to display, and possibly change, LOB and XML data. (long)

Explanation: The user entered a SELECT statement using ZDT/Db2 options 4.1-4.4. The result table for the SELECT statement includes either LOB or XML columns. Display of the result table is not supported, the operation cannot be performed.

User response: To display and possibly change LOB or XML data, use the ZDT/Db2 editor functions (Browse, View or Edit) and specify the Db2® object name that contains the LOB or XML columns.

HFMDB338

XML Columns (short). Edit of objects with XML columns is only supported when the target Db2® system is at Db2® V10 NFM. The function cannot be performed.(long)

Explanation: The user attempted to process a Db2® object containing XML columns using the ZDT/Db2 editor. The target Db2® system is not at Db2® V10 new function mode (or later). Display of the Db2® object is not supported, the operation cannot be performed.

User response: To display and possibly change XML data, the target Db2® system must be at Db2® V10 new function mode, or later.

HFMDB339

Length error *Len* (short). The maximum length of the LOB *Len* has been exceeded. Either reduce the amount of data or cancel out of the LOB editor session. (long).

Explanation: This message is issued when an attempting to save changes made in a Z Data Tools/Db2 LOB editor session. The total length of the LOB data exceeds the maximum size for the LOB column. The changes made in the LOB editor session cannot be saved.

User response: Either reduce the size of the LOB data, or issue the CANCEL command to abandon any changes made. You can use the TE command in the originating editor session to see the maximum length for the LOB. You can use the LOBLEN command in the originating editor session to show the current length for a LOB column.

HFMDB340

Not a LOB column (short). The field reference does not identify a LOB column. (long).

Explanation: The user issued a command to process a LOB column (LOBLEN, LOBBRWS, LOBVIEW or LOBEDIT). The field reference provided does not identify a LOB column. The command cannot be processed.

User response: Check the field reference entered as part of the command and ensure that it identifies a LOB column. Re-issue the command.

HFMDB341

Not a LOB column (short). The column at the cursor position is not a LOB column. (long).

Explanation: The user issued a command to process a LOB column (LOBLEN, LOBBRWS, LOBVIEW or LOBEDIT), using cursor positioning to identify the row and LOB column of interest. The column identified is not a LOB column. The command cannot be processed.

User response: Ensure that the cursor is located within the bounds for a LOB column. Re-issue the command.

HFMDB342

LOB Length = *Length* (short). The length of the LOB data (Row number = *Row number*, REF# = *Field Ref*) is *Length*. (long).

Explanation: The user issued the LOBLEN command. The length of the column identified is *Length*. *Row number* is the row number for the row that includes the identified column. *Field Ref* is the field reference value entered in the LOBLEN command.

User response: None - informational message.

HFMDB343

No LOB columns (short). The Db2® object does not contain any LOB columns. (long).

Explanation: The user issued a command to process a LOB column (LOBLEN, LOBBRWS, LOBVIEW or LOBEDIT), however the Db2® object being processed does not include any LOB columns. The command cannot be processed.

User response: None - informational message.

HFMDB344

LOBVIEW not available (short). LOBVIEW is not available. You may only view a LOB column from a view or edit session (not browse) of the parent table. (long).

Explanation: The user issued the LOBVIEW command from a Z Data Tools/Db2 browse session. The only LOB-related commands that are available in a Z Data Tools/Db2 browse session are LOBLEN and LOBBRWS. The command cannot be processed.

User response: Use the LOBBRWS command instead. If LOBVIEW is required exit the editor session and re-invoke the editor using either the view or edit function.

HFMDB345

LOBEDIT not available (short). LOBEDIT is not available. You may only edit a LOB column from an edit session (not browse or view) of the parent table. (long).

Explanation: The user issued the LOBEDIT command from either a Z Data Tools/Db2 browse, or a ZDT/Db2 view session. The only LOB-related commands that are available in a Z Data Tools/Db2 browse session are LOBLEN and LOBBRWS. The only LOB-related commands that are available in Z Data Tools/Db2 view session are LOBLEN, LOBBRWS and LOBVIEW. The command cannot be processed.

User response: Use one of the available commands as documented in the explanation. If LOBEDIT is required exit the editor session and re-invoke the editor using either the edit function.

HFMDB346

Empty LOB (short). The LOB has no data. (long).

Explanation: This message is issued when the data for a LOB column is initially displayed by the ZDT/Db2 LOB editor. There is no data in the LOB column.

User response: None - informational message.

HFMDB347

Command not supported (short). XML fields cannot be browsed, viewed, or edited under CICS. (long)

Explanation: A command to process an XML column (XMLBRWS, XMLVIEW or XMLEDIT) was issued. These commands are not available when using the ZDT/CICS interface.

User response: None.

HFMDB348

No matching columns (short). The Db2® object contains LOB columns, however there are no selected LOB columns. You can use the TEdit command to change the selection status for a column. (long).

Explanation: The user issued a command to process a LOB column (LOBLEN, LOBBRWS, LOBVIEW or LOBEDIT), however the template for the object has been modified so that all LOB columns are not displayed. The command cannot be processed.

User response: Issue the Tedit command to display the Column Selection/Edit panel (Template editor). Change the Selection status of the LOB column of interest and save the resulting changes. Re-issue the command.

HFMDB349

BLOB edit not supported (short). Edit of a binary large object column is not supported. (long).

Explanation: The LOBEDIT command was issued against a BLOB column. Edit of BLOB data is not supported. The command cannot be processed.

User response: None - informational message.

HFMDB350

Not an XML column (short). The field reference does not identify an XML column. (long).

Explanation: The user issued a command to process an XML column (XMLLEN, XMLBRWS, XMLVIEW or XMLEDIT). The field reference provided does not identify an XML column. The command cannot be processed.

User response: Check the field reference entered as part of the command and ensure that it identifies an XML column. Re-issue the command.

HFMDB351

Not an XML column (short). The column at the cursor position is not an XML column. (long).

Explanation: The user issued a command to process an XML column (XMLLEN, XMLBRWS, XMLVIEW or XMLEDIT), using cursor positioning to identify the row and XML column of interest. The column identified is not an XML column. The command cannot be processed.

User response: Ensure that the cursor is located within the bounds for an XML column. Re-issue the command.

HFMDB352

XML Length = *Length* (short). The length of the XML data (Row number = *Row number*, REF# = *Field Ref*) is *Length*. (long).

Explanation: The user issued the XMLLEN command. The length of the column identified is *Length*. *Row number* is the row number for the row that includes the identified column. *Field Ref* is the field reference value entered in the XMLLEN command.

User response: None - informational message.

HFMDB353

No XML columns (short). The Db2® object does not contain any XML columns. (long).

Explanation: The user issued a command to process an XML column (XMLLEN, XMLBRWS, XMLVIEW or XMLEDIT), however the Db2® object being processed does not include any XML columns. The command cannot be processed.

User response: None - informational message.

HFMDB354

XMLVIEW not available (short). XMLVIEW is not available. You may only view an XML column from a view or edit session (not browse) of the parent table. (long).

Explanation: The user issued the XMLVIEW command from a Z Data Tools/Db2 browse session. The only XML-related commands that are available in Z Data Tools/Db2 browse session are XMLLEN and XMLBRWS. The command cannot be processed.

User response: Use the XMLBRWS command instead. If XMLVIEW is required exit the editor session and re-invoke the editor using either the view or edit function.

HFMDB355

XMLEDIT not available (short). XMLEDIT is not available. You may only edit an XML column from an edit session (not browse or view) of the parent table. (long).

Explanation: The user issued the XMLEDIT command from either a Z Data Tools/Db2 browse, or a ZDT/Db2 view session. The only XML-related commands that are available in a Z Data Tools/Db2 browse session are XMLLEN and XMLBRWS. The only LOB-related commands that are available in a Z Data Tools/Db2 view session are XMLLEN, XMLBRWS and XMLVIEW. The command cannot be processed.

User response: Use one of the available commands as documented in the explanation. If XMLEDIT is required exit the editor session and re-invoke the editor using either the edit function.

HFMDB356

Invalid target column (Short). FMAP cannot be used with BLOB, CLOB, DBCLOB, or XML columns. (Long).

Explanation: The FMAP command was issued against a LOB (CLOB, BLOB, or DBCLOB) or XML column. This is not supported.

User response: None.

HFMDB358

No matching columns (short). The Db2® object contains XML columns, however there are no selected XML columns. You can use the TEdit command to change the selection status for a column. (long).

Explanation: The user issued a command to process an XML column (XMLLEN, XMLBRWS, XMLVIEW or XMLEDIT), however the template for the object has been modified so that all XML columns are not displayed. The command cannot be processed.

User response: Issue the Tedit command to display the Column Selection/Edit panel (Template editor). Change the Selection status of the XML column of interest and save the resulting changes. Re-issue the command.

HFMDB359

Error line *Line* (short). The Db2® XML parser reported an error on or near line *Line*. (long).

Explanation: This message is issued in response to an attempt to save changes made in an XML editor session. There is a syntax error in the XML document that will prevent the updated document being saved in the Db2® object. *Line* indicates the line number where the syntax error is located. The XML document cannot be saved until the syntax error is correcte.

User response: Locate the specified line number, correct the syntax error and attempt to exit the XML edit session again. If it is not possible to resolve the error issue the CANCEL command to abandon any changes made.

HFMDB360

Invalid target column (short). FMAP cannot be used when the column cannot be updated, for example, columns defined with the GENERATED attribute

Explanation: The FMAP editor primary command was issued against a column with the GENERATED attribute. This is not supported.

User response: None.

HFMDB362

Null XML column (short). The XML column has the NULL value (long).

Explanation: The XMLEDIT command was issued against an XML column. The XML column has the Db2® null value.

User response: None.

HFMDB363

Unable to process (short). The object being edited does not include all columns of any unique indexes defined on the base table. ZDT/Db2 is unable to uniquely identify rows in the result table using SQL - this is required for XMLedit functionality (long).

Explanation: An XML edit operation was issued against an XML column. The object being processed is a Db2® view. The columns in the view do not include all columns of any unique indexes defined on the view's base table. In this situation ZDT/Db2 cannot uniquely identify rows in the result table, which is required for the XML editor's functionality. The command cannot be processed.

User response: Retry the operation using the base table to which the view refers.

HFMDB364

Unable to process (short). An attempt to obtain the RID value for the row failed. The RID value is required for XMLedit functionality (long).

Explanation: An XML edit operation was issued against an XML column. The object being processed is a Db2® view. ZDT/Db2 attempted to extract the RID value for the row containing the XML document to be processed. This operation failed. The command cannot be processed.

User response: Retry the operation using the base table to which the view refers.

HFMDB367

LOB column(s) (short). Import/Copy of tables containing LOB (BLOB, CLOB or DBCLOB) columns is not supported (long).

Explanation: An attempt was made to import or copy data containing LOB (BLOB, CLOB or DBCLOB) columns. This is not supported.

User response: None.

HFMDB368

XML column(s) (short). Import/Copy of tables containing XML columns is not supported (long).

Explanation: An attempt was made to import or copy data to a table containing XML columns. This is not supported.

User response: None.

HFMDB369

XML column(s) (short). Tables containing XML columns are not supported (long).

Explanation: An attempt was made to create or export data referencing a table containing XML columns. This is not supported.

User response: None.

HFMDB370

Warning: Saving changes to the SQL statement invalidates any expression previously entered using the "Row Selection Criteria" panel. (Long).

Explanation: The SQL command was issued in the template editor after a condition has been specified using the "Row Selection Criteria" panel. Expressions issued using the SQL command and the "Row Selection Criteria" panel are mutually exclusive, saving an expression entered using the SQL command causes an expressions entered using the "Row Selection Criteria" panel to be discarded.

User response: Informational message. In general, the SQL panel is intended for expressions that cannot be entered using the "Row Selection Criteria" panel, or for users who are comfortable entering native SQL.

HFMDB371

Not allowed. (Short). You cannot de-select columns when creating data using the ZDT/Db2 data create function. When a row is inserted, every column of the object is affected. (Long).

Explanation: An attempt was made to de-select columns in the template for a Db2 object specified in the ZDT/Db2 data create utility. This is not supported.

User response: Ensure that every column of the template is selected.

HFMDB372

De-selected columns (Short). You may not de-select columns when creating data using the ZDT/Db2 data create function. When a row is inserted, every column of the object is affected. (Long).

Explanation: An attempt was made to use a template with de-selected columns with a Db2 object specified in the ZDT/Db2 data create utility. This is not supported.

User response: Ensure that every column of the template is selected.

HFMDB373

Command not available (Short). Commands such as INSERT, REPEAT, COPY, and GET are not available when the editor is operating in large mode. (Long).

Explanation: A command was issued in an ZDT/Db2 editor sessions, operating in Large edit mode, that would have added a row to the editor session. Examples: INSERT, COPY, REPEAT, and GET. This is not supported.

User response: When editing in Large Table mode, you cannot insert new rows.

HFMDB400

Incomplete clause (short). The syntax clause just exited is not yet complete. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function, when the user exits the syntax diagram for a code fragment without making any choice.

User response: No action required. Informational message.

HFMDB401

Command command ignored (short). You entered the *Command* command but the cursor was not on a syntax element. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function when the user presses a PFKey assigned to a command (for example INSRPT, assigned to PF5 by default), however the cursor was not located on an applicable syntax element. The command cannot be performed.

User response: No action required. Informational message.

HFMDB402

Command command ignored (short). You entered the *Command* command with the cursor on a syntax element but the element is not in a repeat group. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function when the user presses a PFKey assigned to a command (for example INSRPT, assigned to PF5 by default), however the cursor was not located on an applicable syntax element. Some commands are only applicable to repeatable syntax elements. The command cannot be performed.

User response: No action required. Informational message.

HFMDB403

No data entered (short). No data was entered for the data entry syntax element. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function when the user exits a data entry syntax element without entering a value. An example is the "table" option of the "table reference clause". A Db2® table name, such as DSN8x10.EMP is required.

User response: No action required. Informational message. If this element is a required part of the SQL syntax you will need to enter a valid value to be able to complete the SQL SELECT statement.

HFMDB404

Statement empty (short). You entered a scroll command but there is no generated statement data. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function when the user enters a scroll command (or presses a PFKey assigned to a scroll command), however there is not data to scroll.

User response: No action required. Informational message.

HFMDB405

End of data (short). The last line of the generated statement is already displayed. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function when the user enters a scroll down command (or presses a PFKey assigned to scroll down), however there the last line of the generated statement is already visible. No further scrolling is possible.

User response: No action required. Informational message.

HFMDB406

Top of data (short). The first line of the generated statement is already displayed. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function when the user enters a scroll up command (or presses a PFKey assigned to scroll up), however there the first line of the generated statement is already visible. No further scrolling is possible.

User response: No action required. Informational message.

HFMDB407

Autoscrolled (short). The generated statement has changed, and the statement generation area has been scrolled to bring the end of the data into view. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function. The SELECT statement being generated changed as a result of user input. The display of the generated statement was automatically scrolled to show the last part of the statement.

User response: No action required. Informational message.

HFMDB408

Statement too long (short). The length of the generated statement exceeds the maximum length allowed for the target system: execution failure expected. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function. The length of the generated SQL statement exceeds 32765 bytes, which is the limit for the function. Execution of the statement is expected to fail and should not be attempted. Note that while Db2® supports SQL statements up to 2M-1 the limit of 32K-2 is imposed by Z Data Tools under ISPF.

User response: Cancel the current operation and restart.

HFMDB409

Repeat separator invalid (short). A repeat separator cannot be entered because no elements of the repeat item have been selected. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function, when the user attempted to add a repeat separator (for example AND, OR), however no repeatable items have been selected.

User response: Select at least one repeatable item before attempting to insert a repeat separator.

HFMDB410

Nothing to execute (short). The EXECUTE command cannot be executed because no statement has been generated for execution. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function. The user entered the EXECUTE command; or pressed the PFKey assigned to EXECUTE, however no SELECT statement has been generated.

User response: Use the prototyping facilities to generate a SELECT statement prior to attempting execution.

HFMDB411

Repeat separator invalid (short). There is no repeat separator associated with the last repeated element. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function. An attempt was made to insert a repeat separator, however there is no repeat separator for the last repeated element.

User response: No action required. Informational message. If in doubt consult the syntax diagram for SELECT in the Db2® SQL Reference manual.

HFMDB412

Nothing to execute (short). The EXECUTE command cannot be executed because no statement has been entered for execution. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute and Explain SQL function. The user entered the EXECUTE command; however the SQL Statement input area is blank.

User response: No action required. Informational message.

HFMDB413

Invalid statement type (short). An invalid statement or a statement that is not dynamically executable was found in the statement entry area. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute and Explain SQL function, when the user has entered an SQL statement that cannot be dynamically prepared/executed. An example is 'CONNECT TO *server*', which can only be coded in an application program. This statement cannot be dynamically prepared and executed. The operation cannot be performed.

User response: Consult the Db2® SQL Reference manual for information about those statements that can/cannot be dynamically prepared/executed.

HFMDB414

Statement type statement executed (SQL cost *Cost*)

Explanation: This message is issued by ZDT/Db2 functions after the successful execution of some SQL statement. *Statement type* is SELECT, INSERT, and so on. *Cost* is the timer on value returned by Db2® when the SQL statement was prepared, giving a relative estimate of cost of executing the statement.

User response: No action required. Informational message.

HFMDB415

count statements executed (SQL cost *Cost*).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function after successful execution of multiple SQL statements. *Count* is the number of SQL statements that were executed. The statements may be of different types. *Cost*is the total of the timer on values returned by Db2® when each SQL statement was prepared.

User response: No action required. Informational message.

HFMDB419

Execution error (short). SQL execution failed. The cursor is placed on the start of the failing statement, or for a syntax error, on the token in error.

Explanation: This message is issued by the ZDT/Db2 Enter, Execute and Explain SQL function when execution of some SQL statement fails. Depending on the type of error, the cursor is placed at the start of the failing SQL statement, or on the token within the SQL statement considered to be in error.

User response: For syntax errors, the cursor location is at the failing token as determined by the Db2® syntax parser. The actual error may be earlier in the statement. Example: SELECT FROM DSN8810.EMP gives SQLCODE-104 and the cursor is located on the period in "DSN8810.EMP". The actual error is the omission of '*' after the SELECT keyword. Consult the Db2® SQL Reference manual for the statement type for the syntax diagram for that statement.

HFMDB420

Data truncated (short). Data found beyond column 72 in the SQL edit session has been discarded. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function after the user exits from an ISPF editor session displayed in response to the SQL command. Within the ISPF editor, session changes may be made to the SQL statement. However, the statement must be entered in columns 1-72, inclusive. Any data entered in columns 73-80 is truncated (discarded).

User response: Examine the SQL statement to determine where the truncation occurred and make corrections.

HFMDB421

SQL too long (short). The SQL data returned from the SQL edit session was too large for the execution buffer. Some lines have been discarded. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute and Explain SQL function after the user exits from an ISPF editor session displayed in response to the SQL command. Not all of the SQL statement entered in the ISPF editor session could be copied to the display area for the ZDT/Db2 Enter, Execute, and Explain SQL function. The statement is too long.

User response: You may be able to circumvent this problem by using less lines to code the SQL statement in the ISPF editor session. For example, by removing some of the white space in the statement.

HFMDB422

Nothing to edit (short). There is no SQL statement for the SQL command to edit. (long).

Explanation: This message is issued by the ZDT/Db2 Advanced SELECT Prototyping function. The user entered the SQL command; however there was no SQL to display in an ISPF editor session. The SQL command cannot by processed.

User response: Enter some part of the SQL SELECT statement before entering the SQL command.

HFMDB423

Select an option (short). Either edit or execute must be selected. (long).

Explanation: This message is issued by the ZDT/Db2 Edit/Execute SQL (Data Set) function. Neither the "Edit data set", nor the "Execute SQL from data set" options have been selected.

User response: You must select one, or both of the "Edit data set" and "Execute SQL from data set" options to use this function.

HFMDB424

Statement type statement *Action*.

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function when an SQL statement other than a SELECT, DELETE, or UPDATE is processed. *Statement type* is the type of statement (Example SET) and *Action* indicates the type of operation. For example, "executed".

User response: No action required. Informational message.

HFMDB425

count statements *Action*

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function when multiple SQL statements are processed. *count* is the type of statement and *Action* indicates the type of operation. For example, "executed".

User response: No action required. Informational message.

HFMDB426

SQL execution failed (*count* statement(s) executed successfully).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function, after multiple SQL statements were processed. The execution of at least one SQL statement failed, however *count* statements were processed successfully.

User response: Examine the list of SQL statements processed to determine whether the failure to execute a statement (or statements) is appropriate.

HFMDB427

No columns selected (short). The EXECUTE command cannot be executed because no columns have been selected from which to retrieve data. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. The user attempted to execute the SQL SELECT statement. However, the SELECT statement is incomplete and cannot be executed. There are no columns specified in the SELECT clause.

User response: You need to specify at least one column in the SELECT statement, for the statement to be executable. Select a column by typing 'S' next to the entry for the column, or use 'ALL' to select all columns.

HFMDB429

Value required (short). A value must be entered to build a where clause predicate. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. The user entered a value in the 'Op' (Operator) column for a column, but no value was entered in the corresponding 'Value' column. A value is required to construct a valid WHERE clause, which has the form: COLUMN *operator value*.

User response: Enter a value appropriate for the column being included in the WHERE clause.

HFMDB431

Invalid operator (short). Enter a valid logical operator: AND, OR, NOT, NOR. NOT represents AND NOT and NOR represents OR NOT. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. The user entered an invalid value in the 'LOp' (Logical Operator) column for a column. The LOp column is used to specify a logical operator. This is used between any WHERE clause predicates specified for the SQL statement being prototyped.

User response: Enter one of the values shown in the long version of the message; the default value is 'AND', selected automatically when no value is specified.

HFMDB432

Invalid operator (short). Enter a valid expression operator: =, ≠,<,>,≤, ≥,IN, NI, LI, NL. LI represents the SQL LIKE operator, NL represents NOT LIKE, and NI represents NOT IN. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. The user entered an invalid value in the 'Op' (Operator) column for a column. The Op column is used to specify a comparison operator. This is used in the WHERE clause predicate being added to the SQL statement being prototyped.

User response: Enter one of the values shown in the long version of the message; the default value is '=', selected automatically when no value is specified.

HFMDB433

Invalid margins (short). The left margin must be less than or equal to the right margin, and both values must be greater than zero. (long).

Explanation:

This message is issued by the ZDT/Db2 Edit/Execute SQL (Data Set) function. The user entered values in the "First column" and "Last column" fields that are invalid. Suggested values are:

- (First column) and 72 (Last column), corresponding to the margins when editing a fixed-block data set with logical record length of 80 and an ISPF profile setting of 'NUMBER ON STD'.
- (First column) and 80 (Last column), corresponding to the margins when editing a fixed-block data set with logical record length of 80 and an ISPF profile setting of 'NUMBER OFF'.

User response:

Consider the recommended values outlined in the explanation. Any values entered should meet the following requirement: 0 < first <=last <=logical record length, where:

- First is the value entered in the "First column" field.
- Last is the value entered in the "Last column" field.
- Logical record length is the LRECL for the data set being edited. You can use ISPF option 3.2 to determine the LRECL for a data set. To enhance the usability of the data set, the value for last should be significantly larger than the value for first.

HFMDB434

Nothing to explain (short). The EXPLAIN command cannot be executed because no statement has been entered for explanation. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function. The user entered the EXPLAIN command. However, there was no SQL statement in the "SQL Statement" area of the panel. The EXPLAIN command requires an SQL statement to process.

User response: Enter an SQL statement in the "SQL Statement" area of the panel and reissue the EXPLAIN command.

HFMDB435

Invalid statement type (short). An invalid statement or a statement that is not explainable was found in the statement entry area. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function. The user entered the EXPLAIN command. However, the SQL statement entered is not "explainable".

User response: Not every SQL statement is "explainable". See the Db2® SQL Reference manual description of the EXPLAIN statement, for a detailed explanation of which SQL statements are explainable. SELECT, MERGE, INSERT, and some types of UPDATE and DELETE statements are considered" explainable".

HFMDB436

Statement executed, no rows deleted.

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function. The user entered a DELETE SQL statement, and the statement was executed. However, Db2® reported that no rows were deleted.

User response: No action required. Informational message.

HFMDB437

Statement executed, no rows updated.

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function. The user entered an UPDATE SQL statement, and the statement was executed. However, Db2® reported that no rows were updated.

User response: No action required. Informational message.

HFMDB438

SQL execution halted (short). Execution of the SQL statement was halted because the user pressed CANCEL on the predictive governor. (long).

Explanation: This message is issued by the ZDT/Db2 Enter, Execute, and Explain SQL function. The execution of an SQL statement was halted because it exceeds values set by the SQL predictive governor. In this situation a warning pop-up panel is displayed, and the user elected to CANCEL execution of the statement.

User response: No action required. Informational message.

HFMDB439

CONNECT stmt. Invalid (short). A CONNECT statement cannot be executed. This statement can only be specified within an application program. (long).

Explanation: This message is issued by the ZDT/Db2 Edit/Execute SQL (Data Set) function. The input data set contains a CONNECT SQL statement. This statement cannot be dynamically prepared and executed. It can only be embedded in an application program. The statement cannot be executed.

User response: Remove the CONNECT statement and retry the operation.

HFMDB440

VIEW (clustered sampling)

Explanation: This message is displayed by the ZDT/Db2 Editor, when using clustered sampling.

User response: No action required. Informational message.

HFMDB441

TABLE (clustered sampling)

Explanation: This message is displayed by the ZDT/Db2 Editor, when using clustered sampling.

User response: No action required. Informational message.

HFMDB442

VIEW (random sampling)

Explanation: This message is displayed by the ZDT/Db2 Editor, when using random sampling.

User response: No action required. Informational message.

HFMDB443

TABLE (random sampling)

Explanation: This message is displayed by the ZDT/Db2 Editor, when using random sampling.

User response: No action required. Informational message.

HFMDB444

LOB

Explanation: This message is displayed by the ZDT/Db2 Editor, when processing LOB data.

User response: No action required. Informational message.

HFMDB445

XML

Explanation: This message is displayed by the ZDT/Db2 Editor, when processing XML data.

User response: No action required. Informational message.

HFMDB450

Copy not allowed (short). The index has not been defined with COPY YES, so it cannot be selected for copying. (long).

Explanation: This message is issued by the ZDT "Db2 Utilities" function. The user selected an index created with the 'COPY NO' option for copy. This is not allowed.

User response: You can only copy a Db2® index when it was originally created with the 'COPY YES' option. See the Db2® Utilities Guide and Reference for more information.

HFMDB451

Chng limits not allowed (short). Change limits cannot be specified because the table space has not been defined with TRACKMOD YES. (long).

Explanation: This message is issued by the ZDT "Db2 Utilities" function. The user entered change limit values for a table space defined with 'TRACKMOD NO'. This is not allowed.

User response: You can only specify change limit values for the Db2® COPY utility when the table space was defined with 'TRACKMOD YES'. See the Db2® Utilities Guide and Reference for more information.

HFMDB453

Cannot recover object (short). Either the Db2® object specified does not exist in the Db2® catalog, or no SYSCOPY entries exist for the Db2® object. (long).

Explanation: This message is issued by the ZDT "Db2 Utilities" function. The user is attempting to recover either a table space, or index. When ZDT/Db2 retrieved a list of entries from SYSIBM.SYSCOPY, no rows were returned. This might occur when wildcard values for database name and table and index space name were specified on the function entry panel.

There are no matching table and index spaces. Another possibility is that there are matching table and index spaces, but no suitable image copies are available for these objects, so the RECOVER utility cannot execute.

User response: No action required. Informational message.

HFMDB454

Invalid change limit (short). The change limit must be an integer value between 0 and 100. (long).

Explanation: This message is issued by the ZDT "Db2 Utilities" function. Change limit values for the COPY utility are expressed as percentage values (0-100 inclusive). The value entered is out of range.

User response: Specify a value in the range 0-100 inclusive. See the Db2® Utilities Guide and Reference for more information.

HFMDB455

Function canceled (short). Load utility processing was canceled by user. (long).

Explanation: This message is issued by the ZDT "Db2 Utilities" function. The CANCEL PFKey or CANCEL command was used when processing for a LOAD utility job, using a Z Data Tools template, was incomplete.

User response: No action required. Informational message.

HFMDB456

Function completed (short). Load utility processing has completed. (long).

Explanation: This message is issued by the ZDT "Db2 Utilities" function. Processing for a Db2® LOAD utility completed normally.

User response: No action required. Informational message.

HFMDB457

Choose at least one item (short). For the given processing option, you must choose at least one item. (long).

Explanation: This message is issued by the ZDT/Db2 "OPTIONS Options" panel. The user selected a processing option of 4 "Options to be included", but did not select any of the available options. At least one option must be selected.

User response: Select one or more of options: 'PREVIEW', 'LISTDEFDD', 'TEMPLATEDD', and 'EVENT'.

HFMDB458

Null indicator mismatch (short). The source Db2® column *Column name* is not nullable and yet the corresponding target data set field is associated with a nullable field. (long).

Explanation: This message is issued by the ZDT/Db2 Unload utility. During mapping between the columns of the source object and fields of the output template, a mismatch for column *Column name*, and the corresponding field, has been detected. *Column name* is defined as NOT NULL. However, the target field has an associated null field indicator.

User response: This usually indicates a mapping problem. The target field for a column defined as "NOT NULL" should not have a null field indicator. Check the target template copybook or template definition and make changes. Retry the operation.

HFMDB459

Enter required field (short). The field where the cursor is positioned is mandatory. (long).

Explanation: This message is issued by the ZDT/Db2 "OPTIONS Options" panel. The user selected a processing option of 3 ('KEY'), but did specify a key value.

User response: Specify a key value in the field indicated by the cursor position.

HFMDB460

Function canceled (short). Unload table utility processing was canceled by user. (long).

Explanation: This message is issued by the ZDT/Db2 Unload utility, when the user presses the PFKey assigned to CANCEL (or types the CANCEL command), prior to completion of the function.

User response: No action required. Informational message.

HFMDB461

Function completed (short). Unload utility processing has completed. (long).

Explanation: This message is issued by the ZDT/Db2 Unload utility, when the function completes normal processing.

User response: No action required. Informational message.

HFMDB462

Settings failure (short). The TEMPLATE, LISTDEF, or OPTIONS settings for the current subsystem may be incorrect. (long).

Explanation: This message is issued during ZDT/Db2 options initialization. An error was encountered when storing values in an ISPF table in the user ISPF profile.

User response: Logoff completely, logon again, and retry the operation. If the problem persists, try deleting all HFM2* members from the user ISPF profile data set. If this fails to resolve the problem contact HCL Support.

HFMDB463

Setting failure (short). When you return from another subsystem, the changes that you made to LISTDEF, TEMPLATE, or OPTIONS settings may not be restored. (long).

Explanation: This message is issued during ZDT/Db2 options initialization. An error was encountered when storing values in an ISPF table in the user's ISPF profile.

User response: Logoff completely, logon again, and retry the operation. If the problem persists, try deleting all HFM2* members from the user ISPF profile data set. If this fails to resolve the problem contact HCL Support.

HFMDB464

Null indicator mismatch (short). A null indicator byte maps to field name *field name* (column *number*), but the name does not contain 'NULL'. (long).

Explanation: This message is issued by the ZDT/Db2 Unload utility. A problem has been detected when mapping columns of the Db2® object to corresponding fields in the Z Data Tools (base) template for the unloaded data.

User response: Refer to the online tutorial topic "Null indicator byte processing", accessed as follows: 3.9.7.1.7 (Utilities > Utilities > Unload Utility > Unload (table) > Null indicator byte processing. Briefly, when using a Z Data Tools Base component template to describe the format for the UNLOAD data set, it is possible to use a field with length > 1 byte for a Db2® null indicator. However, the field name must include the string 'NULL' to be automatically mapped.

HFMDB465

Null indicator mismatch (short). A null indicator byte maps to the byte before the first field. (long).

Explanation: This message is issued by the ZDT/Db2 Unload utility. This is a "should not occur" error.

User response: Check the template for the target (unload) data set is appropriate for the Db2® object being unloaded. Consider using a Z Data Tools/Db2 template generated from the source object to map the unloaded data. If both of these techniques fail to resolve the problem, contact HCL Support.

HFMDB467

Null indicator mismatch (short). The null indicator byte *column name* maps to field *field name* (column *number*) which is mapped already to another table column. (long).

Explanation: This message is issued by the ZDT/Db2 Unload utility. A problem has been detected when processing the mapping for a Db2® null indicator for column *column name*.

User response: Refer to the online tutorial topic "Null indicator byte processing accessed as follows: 3.9.7.1.7 (Utilities > Utilities > Unload Utility > Unload (table) > Null indicator byte processing. Briefly, a field in the template for the unload data set can only be mapped to a single column, or Db2® null indicator.

HFMDB468

Utility service failed (short). ZDT/Db2 initiated an unload table utility operation from export. However, the unload table utility failed to initialize properly. (long).

Explanation: This message is issued when there is a problem generating a JCL deck using ISPF file tailoring. This is a "should not occur" error under normal circumstances.

User response: Contact the person responsible for installing and maintaining ZDT/Db2. Possible causes of this error include setup-related issues such as the SHFMSLIB library not being available to the ZDT/Db2 user.

HFMDB469

Settings failed (short). The TEMPLATE, LISTDEF, or OPTIONS settings (for subsystem *Db2 SSID*.) cannot be loaded or the associated settings could not be found. (long).

Explanation: This message is issued during ZDT/Db2 options processing. Once cause of this error is a missing options module (HFM2POPT), or no definitions for the current Db2® subsystem.

User response: Contact the person who is responsible for installing/maintaining ZDT/Db2. Defining HFM2SSDM macro entries for every Db2® system accessed by ZDT/Db2 is a required installation and customization step. Contact HCL Support if analysis shows no obvious problem in the HFM2POPT module.

HFMDB470

Init install settings (short). The installation-defined settings for this subsystem are used to set options TEMPLATE, LISTDEF, and OPTIONS. (long).

Explanation: This message is issued during ZDT/Db2 Options processing. When processing TEMPLATE, LISTDEF, or OPTIONS, no options had previously been saved in the user's ISPF profile. The option values have been initialized based on any values set in the HFM2POPT (installation) module.

User response: No action required. Informational message.

HFMDB471

Init ZDT/Db2 settings (short). The installation defined setting for this subsystem is: CONNECT = ANY. The ZDT/Db2 defaults are used for options TEMPLATE, LISTDEF, and OPTIONS. (long).

Explanation: This message is issued during ZDT/Db2 Options processing. When processing the connection to a Db2® subsystem, the combination of CONNECT=ANY (HFM2POPI macro parameter) and no HFM2SSDM macro entry for the Db2® subsystem was detected. In this situation there can be no installation settings for TEMPLATE, LISTDEF, and OPTIONS options (since there is no HFM2SSDM macro entry where these settings are defined). The ZDT/Db2 defaults are used to initialize the option values.

User response: No action required. Informational message. The Z Data Tools Customization Guide lists as a required step that an HFM2SSDM macro entry should be defined for every Db2® system that ZDT/Db2 may connect to.

HFMDB472

Init ZDT Db2 settings (short). ZDT Db2 default settings for options TEMPLATE, LISTDEF, and OPTIONS are used. Your installation defined settings uses an old version of the HFM2SDDM macro. (long).

Explanation: This is a "should not occur" error. The HFM2POPT module in use is out-of-date and should be reassembled with the latest macro versions.

User response: Contact the person responsible for installing or maintaining ZDT/Db2. Reassemble the HFM2POPT module using the directions in the *Z Data Tools Customization Guide*. Carefully check the SYSLIB DD statement to ensure that the current version of SHFMSAM1 libraries is in use, and check the SYSLMOD DD statement to ensure that the output module is located in the appropriate library.

HFMDB473

Must provide name (short). A LISTDEF name must be provided when specification LISTDEF is selected. (long).

Explanation: The user selected the 'LISTDEF' specification on the ZDT/Db2 "Db2® Utilities" function entry panel. However, no value was entered in the "LISTDEF name" field. The use of a LISTDEF specification requires a non-blank LISTDEF name.

User response: Enter a valid LISTDEF name, and retry the operation; or change the specification to"Object name".

HFMDB474

//* NULLABLE COLUMN MAPPING

Explanation: This is a message insert used by the ZDT/Db2 Unload utility when preparing JCL.

User response: No action required. This message may be issued when documenting the mapping of nullable Db2® columns in the Db2® object to corresponding fields in the template for the unloaded data.

HFMDB477

//* F (FOOTNOTE) -

Explanation: This is a message insert used by the ZDT/Db2 Unload utility when preparing JCL.

User response: No action required. This message may be issued when documenting the mapping of nullable Db2® columns in the Db2® object to corresponding fields in the template for the unloaded data.

HFMDB478

//* * = FIELD CORRESPONDING TO NULL INDICATOR BYTE IS LARGER

Explanation: This is a message insert used by the ZDT/Db2 Unload utility when preparing JCL.

User response: No action required. This message may be issued when documenting the mapping of nullable Db2® columns in the Db2® object to corresponding fields in the template for the unloaded data.

HFMDB479

//* THAN ONE BYTE. INDICATOR MAPS TO LAST BYTE.

Explanation: This is a message insert used by the ZDT/Db2 Unload utility when preparing JCL.

User response: No action required. This message may be issued when documenting the mapping of nullable Db2® columns in the Db2® object to corresponding fields in the template for the unloaded data.

HFMDB480

WHERE clause too long (short). The WHERE clause is longer than 32767 bytes, and cannot be processed. Specify a shorter WHERE clause. (long).

Explanation: This message may be issued when exiting the ISPF edit session used to specify a WHERE clause in the ZDT/Db2 template editor (Column Selection/Edit panel). The WHERE clause exceeds the maximum allowable length of 32767 bytes, and cannot be processed.

User response: Specify a shorter WHERE clause. Remove any redundant "white space" in the clause and retry the operation.

HFMDB481

Invalid reference (short). The correlation reference *Reference* is invalid. Valid values are #1 - *Maximum reference* inclusive. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. When using an explicit table correlation reference (example #n, n=1, 2 ..) the n must not be greater than the number of Db2® objects entered when the function was first started. "n" must also be a numeric value, example #F is invalid.

User response: Correct the table correlation reference, and retry the operation.

HFMDB482

Invalid column name (short). *Column name* is not a column of the referenced table. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. A qualified column name was entered in the "Value" column, presumably to indicate a join between two columns. The table correlation reference is valid but the column name is not.

User response: Check the value entered for spelling or other errors. Check that the column name is a column of the table indicated by the table correlation reference.

HFMDB483

Ambiguous reference (short). This column name appears in more than one of the selected tables. Use a correlation reference, eg #1.COLNAME to identify the required. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. A value entered in the "Value" column matches more than one column name in the Db2® objects selected for prototyping. A table correlation reference needs to be used to remove any ambiguity.

User response: Precede the column name with a table correlation reference, example #n., where "n" is a positive integer.

HFMDB484

Ambiguous input (short). The value specified is a column name, but can also be interpreted as a constant. To specify a constant enclose the value in quotation marks. To specify a column use a correlation reference, eg #1.COLNAME. (long).

Explanation: This message is issued by the ZDT/Db2 Basic SELECT Prototyping function. A value entered in the "Value" column matches a column name in a Db2® object selected for prototyping. There are two interpretations for the value entered:

1. A comparison between the column and a string constant.
2. A join between the column and the column name entered.

User response: Depending on context, either enclose the value in quotation marks (to indicate the first interpretation shown above) or precede to the column name with a table correlation reference (example #n.) to indicate the second interpretation above.

HFMDB487

NOPAD specified (Short). The DB2 UNLOAD NOPAD option is specified. ZDT/Db2 cannot generate a template that describes the data that will be unloaded. (Long).

Explanation: This message is issued by the ZDT/Db2 DB2 UNLOAD utility when the DB2 NOPAD option is selected. The NOPAD option can result in UNLOAD data with varying lengths, the format for these records cannot be described using a Z Data Tools template.

User response: Informational message. If a template is required for the INLOAD data, do not select the DB2 NOPAD option.

HFMDB489

Member not found (Short). The DBRM member cannot be found in the data set recorded in the Db2 catalog. This can happen when members are deleted from a DBRM library after the bind was executed.

Explanation: An attempt was made to browse the contents of a DBRM member using the ZDT/Db2 object list utility. The member specified in the Db2 catalog table could not be found.

User response: Informational message. This situation can occur if the member containing the DBRM is deleted after the BIND command was issued.

HFMDB490

Parameters have been reset to defaults.

Explanation: This message is issued in response to the RESET command entered on a Z Data Tools/Db2 options panel. The values on the panel have been reset to the default values.

User response: No action required. Informational message.

HFMDB491

RESET command failed.

Explanation: This message is issued in response to the RESET command, entered on a Z Data Tools/Db2 options panel. The RESET operation failed, this is a "should not occur" error.

User response: Exit ZDT/Db2 completely. Start ZDT/Db2 again, and retry the operation. If the problem recurs, check for ISPF-related problems such as a full user ISPF profile data set. If there are no obvious external problems, contact HCL Support.

HFMDB500

Copying...

Explanation: This message is issued by the ZDT/Db2 copy utility to indicate that copy operations are in progress.

User response: No action required. Informational message.

HFMDB501

number of rows rows

Explanation: This message is issued by the ZDT/Db2 copy utility to give an indication of progress for the current copy operation. *number of rows* is the number of rows that have been processed.

User response: No action required. Informational message.

HFMDB502

number of rows(s) copied (short). *Number of rows* row(s) copied from Db2® object *source Db2 object*to *target Db2 object*

Explanation: This message is issued by the ZDT/Db2 copy utility, at the completion of the copy operation, to report the number of rows copied from *source Db2 object* to *target Db2 object*. This message indicates that there were no duplicate key or referential integrity errors encountered.

User response: No action required. Informational message.

HFMDB503

number of rows row(s) copied. *insert string*

Explanation: This message is issued by the ZDT/Db2 copy utility, at the completion of the copy operation, to report the number of rows copied. *insert string* contains additional information, typically messages HFMDB504, HFMDB505, or HFMDB507.

User response: No action required. Informational message.

HFMDB504

count duplicate row(s) ignored.

Explanation: This message is issued by the ZDT/Db2 copy utility, at the completion of the copy operation, to report the number of duplicate row errors that were ignored. This message may be issued as part of message HFMDB503.

User response: No action required. Informational message.

HFMDB505

count RI/Constraint errors.

Explanation: This message is issued by the ZDT/Db2 copy utility, at the completion of the copy operation, to report the number of referential integrity, or constraint errors that were encountered. This message may be issued as part of message HFMDB503.

User response: No action required. Informational message.

HFMDB507

count duplicate row(s) updated.

Explanation: This message is issued by the ZDT/Db2 copy utility, at the completion of the copy operation, to report the number of duplicate rows that were updated. This message may be issued as part of message HFMDB503.

User response: No action required. Informational message.

HFMDB510

Function canceled (short). Copy function was canceled by user. (long).

Explanation: The user terminated the ZDT/Db2 copy function by issuing the CANCEL command, typically by pressing the PF12 PFKey (assigned to CANCEL by default).

User response: No action required.

HFMDB511

Rollback completed (short) Copy of *source Db2 object* to *target Db2 object* was rolled back due to errors.

Explanation: This message is issued by the ZDT/Db2 copy utility when the current copy operation was terminated because of errors. A Db2® ROLLBACK is issued automatically in this situation, and all copy operations are backed out.

User response: Determine why the copy operation was terminated, resolve the error condition, and retry the operation.

HFMDB512

Copy canceled (short) Output object *target Db2 object* cannot be same as input object *source Db2 object*.

Explanation: The same Db2® object name was specified on the "from" and "to" panels of the ZDT/Db2 copy utility. Copy from an object to itself is not supported. The Copy operation was canceled.

User response: Specify different source or target Db2® objects, and retry the operation.

HFMDB519

Audit failure (short) Auditing to SMF failed - function terminated. BPX1SMF RC=*return code* (Dec) REASON CODE=*reason code* (Dec)

Explanation: ZDT/Db2 is set-up to write audit records to SMF for the current function. A failure occurred while attempting to write an audit record to SMF. The BPX1SMF call resulted in return code *return code* and reason code *reason code*, both expressed in decimal. The current ZDT/Db2 function is terminated.

User response:

This error usually indicates a set-up problem. Possible reason for an auditing failure to SMF are:

1. The userid does have READ access to the SAF FACILITY class profile BPX.SMF.
2. The HFM2POPT module does not have a valid number specified for the SMFNO parameter.

Consult the systems programmer for assistance in resolving the cause of the error.

HFMDB520

SQL statement too long (short) The SQL statement is > 32K-2 in length. The maximum length for an SQL statement for Db2® versions 6 or 7 is 32765 characters. The statement cannot be executed. (long)

Explanation: ZDT/Db2 detected an SQL statement greater than the 32K-2 limit, which is applicable to Db2® versions prior to Db2® version 8. The SQL statement cannot be executed.

User response: Simplify the SQL statement, and retry the operation.

HFMDB521

SQL statement too long (short) The SQL statement is > 2M in length. The maximum length for an SQL statement for Db2® versions 8 or later is 2097152 characters. The statement cannot be executed.

Explanation: ZDT/Db2 detected an SQL statement greater than the 2M (2097152) limit, which is applicable to Db2® versions 8, and later. The SQL statement cannot be executed.

User response: Simplify the SQL statement, and retry the operation.

HFMDB600

Global Settings

Explanation: This message appears on some ZDT/Db2 options panels to indicate that the global options are being processed. ZDT/Db2 offers two levels of option settings for most functions. These are:

1. Global options, selected using option 0 from the ZDT/Db2 main menu. These options are saved in the ISPF profile data set for the user in the ISPF PROFILE pool, and persist between each ZDT/Db2 session.
2. Local options, selected from the function entry panel ("Edit options"). These options persist until the user exits the current ZDT/Db2 function, and overrides any Global options that may be in effect. See the *Z Data Tools User's Guide for Db2® Data* for a full explanation.

User response: No action required. Informational message.

HFMDB601

Changes discarded (short) Invalid changes to the default high level qualifier have been discarded. (long)

Explanation: An invalid value for the "default high level qualifier" field was specified, followed by an attempt to leave the panel. The invalid value has been discarded.

User response: Examine the value for "default high level qualifier" and re-enter the desired value. Press Enter to validate the value, prior to leaving the panel.

HFMDB602

Changes discarded (short) Invalid changes to the display width values have been discarded. (long).

Explanation: An invalid value for one of the "Display width for varying columns" fields was entered on the "Object List Options (2 of 2)" panel, followed by an attempt to leave the panel. The invalid value (or values) has been discarded.

User response: Use the field level help for each field (place the cursor on the field, press the HELP key, PF1 by default) to determine the valid values for each field. After entering a new value, press ENTER to validate the values, prior to leaving the panel.

HFMDB757

SQLCODE=*SQLCODE*, *row count* row(s) affected.

Explanation: This is a message insert used in the ZDT/Db2 audit report utility. *SQLCODE* is the SQLCODE associated with an SQL statement, which might have affected 1 or more rows. *row count* shows the number of rows, as reported by Db2®, that were affected by the SQL statement. For SQL statements that affect Db2® objects subject to referential integrity constraints, the row count value does not show rows affected in objects other than the primary object affected by the SQL statement./

User response: No action required.

HFMDB765

Privilege statements: *count*

Explanation: This is a message insert used in the ZDT/Db2 audit report utility. *count* shows the number of SQL statements issued by the ZDT/Db2 function that might affect Db2® privileges, typically GRANT and REVOKE.

User response: No action required.

HFMDB800

count rows exported (short) *count*row(s) exported from Db2® object *source Db2 object* to file *file name*. (long).

Explanation: This message is issued at the end of a Z Data Tools/Db2 export function. The number of rows exported, *count* the Db2® object from which the rows were exported, *source Db2 object* and the output file name that contains the exported data, *file name* are shown.

User response: No action required.

HFMDB801

Export canceled (short) Export function was canceled by the user, or error encountered while processing the copybook. (long).

Explanation: This message is issued by the ZDT/Db2 export utility. The current ZDT/Db2 export function is canceled and the ZDT/Db2 export function entry panel is redisplayed.

User response: If the user pressed the END or CANCEL PFkeys on the export "To" panel, none. If the user pressed ENTER on the export "To" panel, examine any copybook or template specified for the export data set for errors.

HFMDB802

Exporting...

Explanation: This message is issued by the ZDT/Db2 export utility to indicate that export operations are in progress.

User response: No action required. Informational message.

HFMDB803

count rows imported (short) *count* rows imported, from file *input file name* to Db2® object *target Db2 object*. *duplicate count* duplicate rows not imported. *error count* rows in error not imported. *selection count* rows dropped via record selection. *REXX drop count* rows dropped by REXX proc. (long).

Explanation:

This message is issued at the end the ZDT/Db2 import utility, and shows summary statistics for the import operation.

- *count* is the number of rows imported.
- *duplicate count* is the number of duplicate row errors that were encountered. The record that caused the duplicate row error has not been imported.
- *error count* is the number of records that generated errors other than a duplicate row error. The record that caused the error has not been imported.
- *selection count* is the number of records (rows) that were dropped as a result of record selection.
- *REXX drop count* is the number of records (rows) that were dropped because of user-specified processing in the REXX procedure. The same information is presented on the "Import Report" panel. This message is issued when the Update option for duplicate key errors is not selected on the import options panel.

User response: No action required.

HFMDB804

Import canceled (short) Import function was canceled by the user. (long).

Explanation: The user canceled the import operation, by pressing the END or CANCEL PFkey.

User response: No action required. Informational message.

HFMDB805

Importing...

Explanation: This message is issued by the ZDT/Db2 import utility to indicate that import operations are in progress.

User response: No action required. Informational message.

HFMDB806

Rollback completed (short) Import of *import file name* to *target Db2 object* was rolled back due to errors. *count* rows rolled back. *duplicate count* duplicate rows not imported. *updated count* duplicate rows updated. *not updated count* duplicate rows not updated. *error count* rows in error not imported. *drop count* rows dropped by record selection. *REXX drop count* rows dropped by REXX procedure. Note, that PF12/CANCEL on SQL error panel will skip each row in error.

Explanation: This is a message insert, used in the ZDT/Db2 audit report utility.

User response: No action required.

HFMDB807

row count rows created (short) *row count*rows created, in Db2® object *output Db2 object name*. *duplicate count* duplicate rows not created. *error count* in error not created.

Explanation: This message is issued by the ZDT/Db2 data create utility. *row count* is the number of rows that were inserted (created) in the output Db2® object *output Db2 object name*.

- *duplicate count* is the number of rows that could not be inserted because of duplicate key (SQLCODE-803) errors.
- *error count* is the number of rows that could not be inserted because of errors other than duplicate key errors.

User response: No action required. Informational message. If there are multiple duplicate key errors, you can edit the template for those columns that form part of the unique index keys to specify a pattern to be used when creating values for that column.

HFMDB808

Create canceled (short) Create function was canceled by the user.

Explanation: The user canceled the ZDT/Db2 data create function, typically be pressing the PF key assigned to CANCEL.

User response: No action required. Informational message.

HFMDB809

Creating...

Explanation: This message is issued by the ZDT/Db2 data create utility to indicate that data create operations are in progress.

User response: No action required. Informational message.

HFMDB810

No rows created (short) Create of *row count* rows was rolled back due to error. Note that PF12/CANCEL on SQL error panel will skip each row in error.

Explanation: The current ZDT/Db2 data create function has been terminated because of errors. No rows have been created in the output Db2® object because a Db2® ROLLBACK statement was issued, and this backed out all uncommitted changes. Prior to the error condition that caused the function to be terminated, *row count* rows had been inserted into the output Db2® object.

User response: If the function was executed online, the failing operation will have produced an SQL error, displayed to the user in the SQL error pop-up panel. Look up the SQLCODE shown and respond as indicated.

HFMDB811

row count duplicate rows not imported.

Explanation: The ZDT/Db2 import utility, executing in batch mode, did not import *row count* rows because of duplicate key (SQLCODE-803) errors.

User response: This is one of several messages that may be issued by the ZDT/Db2 import utility when run in batch mode. If duplicate key errors were not expected, examine the input data to determine why the errors occurred.

HFMDB812

row count row(s) in error not imported.

Explanation: The ZDT/Db2 import utility, executing in batch mode, did not import *row count* rows because of unspecified SQL errors.

User response: This is one of several messages that may be issued by the ZDT/Db2 import utility when run in batch mode. The presence of this message usually indicates a problem with the input data, which should be examined to determine why the errors occurred.

HFMDB813

row count row(s) dropped via record selection.

Explanation: The ZDT/Db2 import utility, executing in batch mode, did not import *row count* rows because the corresponding input record was dropped as part of the record selection process.

User response: This is one of several messages that may be issued by the ZDT/Db2 import utility when run in batch mode. When record selection criteria, or a REXX exec, is specified for the input template, this message indicates that one or more records did not meet the record selection criteria, or were dropped as part of the REXX procedure processing. If no REXX procedure was specified this message may indicate a mismatch between the input data and its associated template.

HFMDB814

row count row(s) dropped via REXX proc.

Explanation: The ZDT/Db2 import utility, executing in batch mode, did not import *row count* rows because the corresponding input record was dropped as part of the associated REXX procedure processing.

User response: This is one of several messages that may be issued by the ZDT/Db2 import utility when run in batch mode. The user response depends on context. It may be reasonable for records to be dropped. If, however, records should not be dropped as part of the REXX procedure processing, examine the input data to determine why the records were dropped.

HFMDB815

row count duplicate row(s) not created.

Explanation: The ZDT/Db2 data create utility, executing in batch mode, did not create *row count* rows because Db2® raised a duplicate key error (SQLCODE-803) when ZDT/Db2 attempted to insert the newly created row in the target Db2® object.

User response: This is one of several messages that may be issued by the ZDT/Db2 data create utility when run in batch mode. The user response depends on context. Db2® will always reject rows that violate uniqueness constraints. When many duplicate row errors are received consider adjusting the input parameters to avoid duplicate values for any index columns.

HFMDB816

row count row(s) in error not created.

Explanation: The ZDT/Db2 data create utility, executing in batch mode, did not create *row count* rows because Db2® raised an error (other than SQLCODE-803) when ZDT/Db2 attempted to insert the newly created row in the target Db2® object.

User response: This is one of several messages that may be issued by the ZDT/Db2 data create utility when run in batch mode. This message usually indicates an error. Db2® may reject the insert of a row for many reasons. Note, that the SQL error issued by Db2® is NOT SQLCODE-803. Duplicate row errors are reported separately. Examples of error SQLCODEs that might result in this message include: -530, -531, and -532 (referential integrity errors).

HFMDB817

row count existing rows(s) updated.

Explanation: The ZDT/Db2 import utility, executing in batch mode, updated *row count* rows. Only columns that are not part of any unique index key are updated.

User response: This is one of several messages that may be issued by the ZDT/Db2 import utility, when run in batch mode. The user response depends on context. In most cases, the update of existing rows is best. If, however, the import operation is not expected to cause duplicate key errors (SQLCODE-803), compare the input records and Db2® object, to discover why there are duplicates.

HFMDB818

row count existing rows(s) not updated.

Explanation: The ZDT/Db2 import utility, executing in batch mode, attempted to update existing rows after the insert operation for the record failed with SQLCODE-803 (duplicate key error). The update operation failed.

User response: This is one of several messages that may be issued by the ZDT/Db2 import utility, when run in batch mode. This message usually indicates an error. Db2® may reject the update of a row for many reasons. Examples of error SQLCODEs that might result in this message include -530, -531, and -532 (referential integrity errors).

HFMDB819

row count import/updt(d) (short) *row count* rows imported, from file *input data set name* to Db2® object *target Db2 object name*. *duplicate row count* duplicate rows updated. *duplicate error count* duplicate rows not updated. *error row count* rows in error not imported. *dropped record count* rows dropped by record selection. *Rexx drop count* rows) dropped by REXX procedure.

Explanation: This message is issued by the ZDT/Db2 import utility, when executed online. It summarises the statistics for the import operation just completed. This message is issued when the "update existing rows" option is selected, and when no rows were inserted, or updated. *row count* is the number of rows imported or updated. In this case, 0.

- *duplicate row count* is the number of records that generated SQLCODE-803 (duplicate key) errors.
- *duplicate error count* is the number of records for which the update of an existing row failed.
- *error row count* is the number of records that generated errors other than SQLCODE-803 for the INSERT operation.
- *dropped record count* is the number of records that were not considered (dropped) as a result of record selection.
- *Rexx drop count* is the number of records that were dropped via the user-specified REXX procedure.

User response: This message is issued when no rows were inserted, and no rows were updated. The user response depends on context. If insert or updates were expected, examine the input data to determine why no records were processed.

HFMDB820

Import complete. *Row count* row(s) imported.

Explanation: This message is issued by the ZDT/Db2 import utility at the completion of the import process. Row count rows were successfully imported.

User response: Informational message.

HFMDB821

Rollback complete. *Row count* row(s) rolled back.

Explanation: This message is issued by the ZDT/Db2 import utility at the completion of the import process. One or more rows were not successfully imported. The import for Row count rows was rolled back.

User response: Informational message.

HFMDB822

Row count rows imported (Short). *Row count* rows imported, from file *input data set name* to Db2 object *output object name*. *duplicate row count* duplicate rows updated. *Duplicate error count* duplicate rows not updated. *error row count* rows in error not imported. *dropped record count* rows dropped by record selection. *Rexx drop count* rows dropped by REXX procedure (Long).

Explanation: Upon completion of the ZDT/Db2 import (DBI), this message is displayed showing the statistics when rows were inserted into the Db2 object.

User response: Informational message.

HFMDB823

update row count rows updated (Short). *update row count* duplicate rows imported from file *input data set name* and updated in object *output object name*. *duplicate error count* duplicate rows not updated. *row count* rows imported. *error row count* rows in error not imported. *dropped record count* rows dropped by record selection. *Rexx drop count* rows dropped by REXX procedure (Long).

Explanation: Upon completion of the ZDT/Db2 import (DBI), this message is displayed showing the statistics when rows were updated in the Db2 object.

User response: Informational message.

HFMDB824

Rows imported/updated (Short).*row count* rows imported from file *input data set name* to Db2 object *output object name*. *duplicate row count* duplicate rows updated. *duplicate error count* duplicate rows not updated. *error row count* rows in error not imported. *dropped record count* rows dropped by record selection. *Rexx drop count* rows dropped by REXX procedure (Long).

Explanation: Upon completion of the ZDT/Db2 import (DBI), this message is displayed showing the statistics when rows were inserted or updated in the Db2 object.

User response: Informational message.

HFMDB825

duplicate row count duplicate row errors.

Explanation: Row count of duplicate rows in error.

User response: Informational message.

HFMDB826

All columns selected (short) No columns were selected in the template for *Db2 object name*. Processing cannot proceed without at least one selected column. ZDT/Db2 has automatically selected all columns for processing.

Explanation: This message is issued by either the ZDT/Db2 export, or ZDT/Db2 UNLOAD utilities. The template editor was used to deselect all columns, followed by an attempt to execute the function with the modified template. Since a Z Data Tools/Db2 function requires at least one selected column to proceed, all columns are automatically selected and this message is issued.

User response: No action required. Informational message. By default, all columns of a Db2® object are selected for processing. While it is valid to deselect some of the columns in the Db2® object, it is invalid to deselect all columns, since this results in a "no-operation" condition.

HFMDB827

Import to view failed (short) The import operation has been canceled. Import to a view is only supported if the view is defined directly on a table. Repeat the operation using the actual table that the view refers to.

Explanation: The ZDT/Db2 import utility was used to import data into a Db2® view. The Db2® view refers to more than one Db2® table, or refers to one or more views. Import to this type of view is not supported. The import operation is canceled.

User response: In general, import to a Db2® view defined as "SELECT * FROM *Db2 table name*" is supported. Import to views that refer to multiple Db2® tables, or views, is not supported. Where possible, use the ZDT/Db2 import utility with a table name, rather than a dependent view.

HFMDB828

Remapping required (short) No columns in the target table are mapped to any fields in the input template. This can occur if the column names in the target object are different to the field names in the input template. Edit the template for the target object to set the mapping between columns/fields in the target/input templates.

Explanation: The ZDT/Db2 import utility was used with an input template that has field names that do not match any of the columns in the target Db2® object. The automatic field mapping process used by ZDT/Db2, cannot reliably determine which fields in the input template correspond with columns in the target Db2® object. Manual mapping of each field in the input template to a column in the target Db2® object is required.

User response: This message may be issued in the situation where the wrong input file template is specified for a Z Data Tools/Db2 import function. Advanced usage of the ZDT/Db2 import utility allows individual mapping of each column in the target Db2® object to a corresponding field of the input template. The column and field names need not match, nor the relative positions of the columns in the Db2® object and fields in the input record. If this is the intended usage, use the template editor ("edit template mapping") to set the relationship between columns and fields, and retry the operation.

HFMDB829

Invalid value (short) The valid values are ALL or a number in the range 0-2147483647. (long).

Explanation: An invalid value was entered in the "Max duplicates" field of the "Copy Options" panel. The valid values are "ALL"or an integer in the range 0-2147483647, inclusive.

User response: Correct the invalid value.

HFMDB831

Browse substituted (short) The Db2® result table is not updateable, specifically because not all columns of the first table named in the from clause are present, nor are all columns of any unique index(es) defined on this table. Browse has been substituted for edit. (long).

Explanation: A SELECT statement was entered using one of the ZDT/Db2 functions that allow free-form SQL statements to be processed. When ZDT/Db2 uses the editor to display the result table from SELECT statement, edit operations may not be possible for free-form SQL statements. This message occurs when the conditions listed in the message are not met. For tables with unique indexes defined, ZDT/Db2 can use the values in the columns that comprise the unique index to uniquely identify any row of the table. However, this requires that every column in the unique index key is represented in the result table. If reliable edit is not possible, ZDT/Db2 converts the edit session to browse to avoid any possibility of data corruption.

User response: No action required. Informational message. If edit operations of the result table are required, the preferred method is to edit the Db2® object directly.

HFMDB832

Update not possible (short) ZDT/Db2 cannot update the data in the row(s) marked with =INERR. This error may occur attempting to edit the result table from an arbitrary select statement entered using option 4.2, 4.3 or 4.4. You should cancel the current edit session and then edit the underlying Db2® object directly.

Explanation: An attempt was made to update data in the row of a result table for an SQL statement entered using ZDT/Db2 options 4.2, 4.3, or 4.4. The update failed because Db2® rejected the update. There are various reasons why this error may occur, including editing the result table for a join. The update operation cannot be performed.

User response: If edit operations of the result table are required, the preferred method is to edit the Db2® object directly.

HFMDB833

Delete not possible (short) ZDT/Db2 cannot delete the data in the row(s) marked with =INERR. This error may occur attempting to edit the result table from an arbitrary select statement entered using option 4.2, 4.3 or 4.4. You should cancel the current edit session and then edit the underlying Db2® object directly.

Explanation: An attempt was made to delete a row of a result table for an SQL statement entered using ZDT/Db2 options 4.2, 4.3, or 4.4. The update failed because Db2® rejected the delete. There are various reasons why this error may occur, including editing the result table for a join. The update operation cannot be performed.

User response: If edit operations of the result table are required, the preferred method is to edit the Db2® object directly.

HFMDB835

Parameter marker invalid (short) The SELECT statement contains one or more "?". Db2® interprets these as parameter markers, resulting in an SQL error. Replace the "?" with constants and rerun the statement.

Explanation: A SELECT statement was entered using ZDT/Db2 options 4.3, or 4.4, and possibly other options. The SELECT statement includes Db2® parameter markers ('?'). While the use of Db2® parameter markers is valid when using, for example, an application programming language, the use of Db2® parameter markers in SELECT statements processed by ZDT/Db2 is invalid.

User response: Locate the Db2® parameter markers in the SELECT statement. Replace each occurrence with a constant of the appropriate type. Rerun the statement.

HFMDB836

Browse substituted (short) This version of ZDT/Db2 does not support new Db2® data types introduced with Db2® version 9. Browse has been substituted for edit. (long).

Explanation: ZDT/Db2 detected a data type in the Db2® object being edited that is not currently supported. To avoid the possibility of data corruption, ZDT/Db2 has substituted browse for edit.

User response: Determine which column has the unsupported data type. Contact HCL Support for information about later versions of ZDT/Db2 which support the new data type.

HFMDB837

XML column(s) (short) This version of ZDT/Db2 does not support the XML data type introduced with Db2® version 9. The function cannot be performed.

Explanation: ZDT/Db2 detected one or more XML columns in the object being processed by the current ZDT/Db2 function (editor, copy, import, export, and print). XML columns are not supported. The function cannot be performed, and processing is terminated.

User response: Specify a Db2® object that does not include XML columns.

HFMDB838

Duplicate count reached (short) The specified duplicate count was reached. The import has been canceled. (long).

Explanation: A Z Data Tools/Db2 import was specified with a non-zero "Max duplicate" count. During the processing of the input data, the number of duplicate row errors reached the specified "Max duplicate" value. The import operation if canceled.

User response: The "Max duplicate" count value is specified on the Import Options panel, selected using the "edit options" field on the Import function entry panel. The response depends on the context of when duplicate row errors are expected. Rerun the import after increasing the "Max duplicate" value. When duplicate row errors are not expected, examine the input data to determine why duplicate row errors are occurring.

HFMDB839

Unable to process (short) The Db2® object contains one or more LOB columns. There is insufficient memory available for ZDT/Db2 to process data from this object, because of the large size of the LOB column(s). Refer to the Edit Tutorial, General information, Restrictions for additional information on how to avoid this problem.

Explanation: ZDT/Db2 does not support the processing of Db2® objects with LOB columns. The function cannot be performed and processing is terminated.

User response: Specify a Db2® object that does not include LOB columns.

HFMDB840

Format not for template (short) The Data Format is not compatible with that used to create the template. Template is *template type*.

Explanation: This message is issued by the ZDT/Db2 import utility when there is a mismatch between the template describing the input data, and the data format option for the import data set. The import cannot be processed, and the import is canceled. The type of template specified is *template type*, which will be one of the following: Db2® Unload; Db2® DSNTIAUL; User defined; Base; ZDT/Db2 SQLDA; Other. (long).

User response: The data format for the input data set is specified on the Import options panel, selected using the "edit options" field on the Import function entry panel. The value specified must match the actual data format of the input data. It must also match the type of template specified in the template, or copybook, specified on the import function entry panel. Verify that the specified template correctly describes the input data. Determine the data format of the input data, use that information to set the appropriate data format value on the Import options panel, and retry the import operation.

HFMDB849

Incompatible options (short) The data format must be 2 (Db2® unload format) when execution option 3 (Batch, using Db2® UNLOAD) is selected. (long).

Explanation: The ZDT/Db2 export function can be used to generate a Db2® utility job that will run the Db2® UNLOAD utility for a Db2® table. When this option is selected, the format of the output data is determined by Db2®. Therefore, the data format must be set to '2' (Db2® unload format). The data format for the output data set is specified on the first export options panel, selected using the "edit options" field on the Export function entry panel

User response: Adjust the data format option as indicated, and retry the operation. Note, that it is also possible to use the ZDT/Db2 export function to export data in "Db2® unload format", without running a Db2® batch job. If this is what is intended, set the execution option on the first export options panel to either '1' (online) or '2' (batch), and retry the operation.

HFMDB850

Must be sequential file (short) The Db2® UNLOAD utility does not support unload of data to a partitioned data set. (long).

Explanation: The ZDT/Db2 export utility was executed with an execution option of '3' (Batch, using Db2® unload). The output data set specified on the export 'To' panel, is a partitioned data set includes a member name. While the ZDT/Db2 export utility can export data in Db2® unload format to a member of a partitioned data set, this is not supported by the Db2® UNLOAD utility.

User response: Specify a sequential file (Data set organization of PS), rather than a partitioned data set (Data set organization of PO), for the output data set. Retry the operation.

HFMDB851

Must be sequential file (short) The Db2® UNLOAD utility requires that the unload data set be a sequential file (DSORG=PS).

Explanation: The ZDT/Db2 export utility was executed with an execution option of '3' (Batch, using Db2® unload). The output data set specified on the export 'To' panel, is not a sequential data set. While the ZDT/Db2 export utility can export data in Db2® unload format to non-sequential data sets, the only data set type supported by the Db2® UNLOAD utility is sequential.

User response: Check the type for the output data set specified for the Db2® unload operation on the ZDT/Db2 export 'To' panel. Ensure that a sequential file (Data set organization of PS) is specified. Retry the operation.

HFMDB852

Not required (short) A query number value is only required with the EXPLAIN primary command. (long).

Explanation: An SQL statement and query number value were entered on the "Enter, Execute and Explain SQL" panel (ZDT/Db2 option 4.3). A query number is only required when the EXPLAIN primary command is issued.

User response: Remove the query number and retry the operation. If the intention was to EXPLAIN the SQL statement, type 'EXPLAIN' on the command line, and press Enter.

HFMDB853

Delete failed (short) The deletion of all rows from the Db2® object failed - the import function has been canceled. (long).

Explanation: The "Delete existing rows" option was specified for a Z Data Tools/Db2 import. When ZDT/Db2 attempted to delete all rows from the target Db2® object, Db2® rejected the operation. The import cannot proceed and the import is canceled. No data has been imported.

User response: There are many reasons why Db2® might reject a 'DELETE * FROM *object name*' statement, including referential integrity errors, locking problems, and unavailable resources. Retry the operation. If the problem persists, issue the 'DELETE * FROM *object name*' directly using ZDT/Db2 options 4.3, 4.4, or Db2® SPUFI. If the operation fails, use the SQLCODE to determine the cause of the failure.

HFMDB855

row count rows+hdr exported (short) *Row count*row(s) and column header row exported from Db2® object *object name*to file *data set name*. (long).

Explanation: This message is issued by the ZDT/Db2 export function and is specific for an export with CSV format and the "Include column headers" option selected. ZDT/Db2 exported *row count* rows from Db2® object *object name* to data set *data*

set name. The first row exported is a header row, containing the names of the Db2® columns in place of the data for those columns.

User response: No action required. Informational message.

HFMDB856

Invalid character ' or " (short) The Delimited Variables (CSV) separator cannot contain either a single quote (') or double quote ("). Choose another value. (long).

Explanation: The user entered a value in the "Separator character" field on the second ZDT/Db2 Export options panel. This field is used to enter a one or two character value for the separator character in "delimited variables (CSV)" export format. The value entered contains either a single or double quote, which is not supported.

User response: The delimited variables (CSV, comma separated variables) format is loosely defined. In situations where the default separator character (comma) is inappropriate, ZDT/Db2 allows the specification of a one, or two, character alternative value. The separator value cannot include either a single quotation mark or double quotation mark, because this causes syntactical ambiguity when the exported data also includes single or double quotation marks, either within the data, or inserted to encapsulate the data. Select another value.

HFMDB858

Must be numeric, >0 (short) The value entered must be an integer in the range 1-2147483647. (long).

Explanation: A value entered in the "start position" field on the editor function entry panel, or the "create count" field on the data create function entry panel, is invalid. A numeric value is required, which must be a positive integer in the range 1-2147483647, inclusive.

User response: Correct the value, retry the operation.

HFMDB859

Must be numeric, >=0 (short) The value entered must be an integer in the range 0-2147483647. You can specify 0, * or ALL to indicate all rows. (long).

Explanation: The value entered in a field on a Z Data Tools/Db2 function entry panel is invalid. A numeric value is required, which must be a non-negative integer in the range 0-2147483647, inclusive. This message is issued for various "row count" input fields. A value of * and ALL can also be entered as synonyms for 0. These values indicate that there is no limit to the number of rows or records to be processed.

User response: Correct the value, retry the operation.

HFMDB890

Db2® subsystem *Db2 SSID* not defined to z/OS®?.

Explanation: This message is issued by a Z Data Tools/Db2 batch job, when connection to Db2® system *Db2 SSID* failed. One reason for this message being issued is that *D2 SSID* does not exist (has not been defined to the local z/OS® system). This might occur in sites running multiple LPARs with shared DASD, where Db2® systems are available on some,but not other, z/OS® systems.

User response: Check that the Db2® SSID specified as a parameter is defined to the z/OS® system where the batch job was executed. If this is not the problem, look up Db2® error code '00F30006' in the Db2® Codes manual and proceed as directed for that error.

HFMDB900

Own authorization ID (short) An authorization ID cannot grant a privilege to itself. (long).

Explanation: This message is issued by the ZDT/Db2 privileges function, in the situation where "An authorization ID or a role attempted to execute a GRANT statement in which that ID itself appears as one of the entries in the list of 'grantee' authorization IDs or role names".

User response: Respond as for SQLCODE-554. The usual remedy is to remove the user current Db2® authid/role from the list of target Db2® authids/roles.

HFMDB903

Identifier too long (short) The identifier is too long. See the SQL Reference for information about the maximum lengths for identifiers. Db2® returned an SQL code of -107 for SQL statement *SQL statement*. (long).

Explanation: This message is issued when an SQL statement contains an identifier that is too long. When connected to Db2® V8 systems or later, this error is unlikely, given the large maximum lengths for most Db2® identifiers, compared to the corresponding Db2® V7 values.

User response: Respond as for SQLCODE-107

HFMDB904

Not authorized (short) Your authorization ID does not have the authority to change the privileges for this object. Db2® returned an SQL code of -551 for SQL statement *SL statement*long)

Explanation: This message is issued by the ZDT/Db2 Object Privileges utility. The Db2® authorization IDs for the current user do not have the authority to change (GRANT or REVOKE) for the object.

User response: Respond as for SQLCODE-551. See your Db2® System Administrator.

HFMDB905

Object not found (short) The specified Db2® object does not exist. Db2® returned an SQL code of -204 for the SQL statement *SL statement* long)

Explanation: This message is issued by the ZDT/Db2 Object Privileges utility. The Db2® object for which the privileges are to be changed could not be found (SQLCODE-204).

User response: Respond as for SQLCODE-204. Check the Db2® object name.

HFMDB907

Cannot be revoked (short) You cannot revoke the specified privilege because the revokee does not possess the privilege or because the specified Db2® authid did not grant the privilege. Db2® returned an SQL code of -556 for SQL statement *SL Statement*. (long).

Explanation: This message is issued by the ZDT/Db2 Object Privileges utility. The privilege cannot be revoked.

User response: Respond as for SQLCODE-556. Refer to the SQL Codes manual, for the appropriate version of Db2®, for a detailed list of possible causes of this error.

HFMDB909

Not authorized (short) Your authorization ID does not have the privilege to issue a REVOKE statement with the BY clause. Db2® returned an SQL code of -552 for SQL statement *SL statement*.long

Explanation: This message is issued by the ZDT/Db2 Object Privileges utility. The privilege cannot be revoked.

User response: Respond as for SQLCODE-552.

HFMDB910

Privilege conflict (short) One or more privileges conflict with one or more other privileges. (long).

Explanation: This message is issued by the ZDT/Db2 Object Privileges utility. An example that causes this error is selecting the 'ALL' option for a privilege, as well as one or more of the individual privileges that are included when 'ALL' is specified.

User response: Correct the specification of the privileges to be processed. In general selecting the 'ALL' option means that you should deselect all of the privileges that are included when 'ALL' is specified.

HFMDB911

Not required (short) This field should be blank for the specified option. (long).

Explanation: On the "Alter Table - VALIDPROC" panel, the "type of VALIDPROC" is set to 1, and a value is also specified for "program name". A value for "program name" is only required when the "type of VALID-PROC" is set to 2.

User response: Check the "type of VALIDPROC" specification. If NULL is the intended specification, remove the value from the "program name" field, and retry the operation. If the intention is to add a program name, change the "type of VALIDPROC" specification to 2, and retry the operation.

HFMDB913

Invalid option (short) The selected option is not one of the listed values or the object type is not supported on the remote Db2® system you specified. This error can occur when you try to list certain Db2® objects on a remote Db2® system that is at an earlier release of Db2® than the system to which you are connected.

Explanation: This message is issued by the ZDT/Db2 Object Privileges utility. The value entered in the "object type" field is invalid. The request cannot be processed.

User response: The valid options for the "Object type" field are listed on the panel. Specify one of the values listed (1-15 inclusive).

HFMDB914

Different Db2® release (short) The release level of the remote Db2® system is different than the release level of the Db2® system to which you are connected. The choice of Db2® object types has been refreshed to show those object types supported on the remote system. Press ENTER again to get the requested list. (long).

Explanation: This message is issued by the ZDT/Db2 Object List and Object Privileges utilities when the user enters a remote location value. ZDT/Db2 connects to the specified location and compares the Db2® version of the remote location against the version of the currently connected Db2® system. When the values are different this message is issued.

User response: No action required. Informational message. Press enter to refresh the list of available object types.

HFMDB915

Empty table (short) The Db2® object contains no rows, or no rows match the specified row selection criteria. (long).

Explanation: This message is issued by the ZDT/Db2 editor when the result table for an object or select statement contains no rows.

User response: No action required. Informational message.

HFMDB916

Excluded rows shadow is off. There are no unexcluded rows to display. (long).

Explanation: In a Z Data Tools/Db2 editor session, the shadow-line for excluded-rows has been turned off, and there are no rows to display. Note, that when the excluded-row shadow-line has been turned off, the string 'SHAD' appears immediately to the left of "Format", at the top right of the display, in TABL mode. For those familiar with ISPF shadow lines, the ZDT/Db2 editor allows the display of the shadow-line indicator to be suppressed, using a command like "SHADOW X OFF". This is not supported in the ISPF editor. Excluded lines are always indicated with a shadow-line. When excluded-row shadow-lines are turned off and when all of the rows in the current editor session are shadow-lines, the editor display appears as though there are no rows in the table (no shadow-line indicator). This message is displayed to alert the user to the fact that there are rows in the result table, even though there are no lines displayed.

User response: No action required. Informational message. You can turn on the shadow-line indicator with "SHADOW X ON".

HFMDB917

No rows to display.

Explanation: In a Z Data Tools/Db2 editor session, the shadow-line for excluded-rows is turned on, and there are no rows to display. See also the message description for HFMDB916. This message is typically issued when edit operations have deleted all rows from the current editor session.

User response: No action required. Informational message.

HFMDB918

Excluded rows shadow is off. Either no unexcluded rows to display, or no rows match the filter specification. (long).

Explanation: In a Z Data Tools/Db2 object list/privileges display, the shadow-line for excluded-rows has been turned off, and there are no rows to display. Note, that when the excluded-row shadow line has been turned off the string 'SHAD' appears immediately to the left of "Format", at the top right of the display, in TABL mode. The ZDT/Db2 object list/privileges utilities use the ZDT/Db2 editor to display lists of objects and privileges. When a filter value is specified for any column, the editor display is adjusted to show only those rows that match the filter value for that column. Any nonmatching rows are converted to excluded-rows, and these rows do NOT produce a shadow-line. This occurs regardless of the current setting for the excluded-row shadow-line. For those familiar with ISPF shadow-lines, the ZDT/Db2 editor allows the display of the shadow-line indicator to be suppressed, using a command like "SHADOW X OFF". This is not supported in the ISPF editor. Excluded-lines are always indicated with a shadow-line. When all of the rows in the current display are shadow-lines, the editor display appears as though there are no rows object list/privilege display (no shadow-line indicator). This message is displayed to alert the user to the fact that there are rows in the object list/privilege display, even though there are no lines displayed. This message is displayed when either:

- A filter value entered for a column results in no matches, or;
- When all rows are excluded.

User response: No action required. Informational message.

HFMDB919

No rows to display, or no rows match the filter specification. (long).

Explanation: In a Z Data Tools/Db2 object/privileges display, the shadow-line for excluded-rows is turned on, and there are no rows to display. The ZDT/Db2 object list/privileges utilities use the ZDT/Db2 editor to display lists of objects and privileges. See also the message description for HFMDB918 on page 1152. This message is displayed when a filter value entered for a column results in no matches.

User response: No action required. Informational message.

HFMDB921

No rows match key (short) There are no rows in the parent table that match the foreign key value. This can occur when a foreign key column(s) has the NULL value. (long).

Explanation: A related edit session of a parent table has no rows that match a foreign key in a dependent table.

User response: No action required. As noted in the long message text, this condition can occur when a foreign key value in the dependent table is NULL.

HFMDB922

No rows match key (short) There are no rows in the dependent table that match the primary/parent key value. (long).

Explanation: A related edit session of a dependent table has no rows with a foreign key that match the primary or parent key value in the parent table.

User response: No action required. A parent/primary key value may exist for which there are no corresponding rows (in terms of foreign key value) in the dependent table.

HFMDB923

Already granted warning (short) At least one of the authid(s) or role(s) already possesses the privilege(s) from the grantor.

Explanation: This message is issued by the ZDT/Db2 Object privileges utility when a GRANT statement returns an SQLCODE +562. This is a warning condition indicating that the GRANT has already been executed for one or more privilege/authid/role combination. The execution of the SQL statement is successful. Any privileges previously held by the target Db2® authids/roles from the grantor are retained (not changed). Any new privileges specified have been successfully GRANTed.

User response: No action required. When there are multiple Db2® authids or roles as target for the GRANT, or multiple privileges being GRANTed, it is not possible to determine which privileges were previously available.

HFMDB924

Grant option warning (short) The WITH GRANT OPTION was ignored for one or more of the privilege(s). (long).

Explanation: A GRANT statement generated by the ZDT/Db2 privileges utility was executed. The statement included the "WITH GRANT OPTION' clause. Db2® returned SQLCODE+558. This is a warning condition indicating that the "WITH GRANT OPTION" clause is redundant. The GRANT statement was successfully executed. An example of an operation that results in this warning condition is attempting to grant the SYSADM privilege to a Db2® authid, with "WITH GRANT OPTION". The

ZDT/Db2 privileges Grant panels allow either 'Y', or 'G', to be specified against the various Db2® privileges; 'Y' corresponds to "GRANT" while 'G' corresponds to "GRANT" and "WITH GRANT OPTION".

User response: No action required.

HFMDB925

Not authorized (short) Your authorization ID does not have the authority to *GRANT|REVOKE* the *privilege type* system privilege. Db2® returned an SQL code of -552 for SQL statement *SL statement*.

Explanation: The ZDT/Db2 privileges utility attempted to execute SQL statement *SL statement*. Execution failed with Db2® returning SQLCODE -552, which indicates that the user does not have the authority to execute the statement.

User response: This is an error condition. Consult with your Db2® security administrator to obtain the necessary Db2® authority to execute the statement.

HFMDB926

Own authorization ID (short) An authorization ID cannot revoke a privilege from itself. (long).

Explanation: The ZDT/Db2 privileges utility was used to issue a REVOKE statement. Execution of the statement failed with Db2® returning SQLCODE-555. This error is issued when the REVOKE statement would have the effect of removing a privilege from a Db2® authid or role that is possessed by the same Db2® authid or role. This is not supported.

User response: Examine the grantees and remove the Db2® authid or role that corresponds to the current Db2® authid or role of the user.

HFMDB927

Invalid name (short) The name is invalid. A database/table space/plan name must be an SQL ordinary identifier - no lower case characters. See the SQL Reference manual for a definition. Db2® returned an SQL code of -113 for SQL statement *SQL statement*.

Explanation: A database/table space/plan name entered using the ZDT/Db2 object privileges utility is invalid. An SQL ordinary identifier is required. See the SQL Reference for a definition. An ordinary identifier is an uppercase letter, followed by zero or more characters, each of which is an uppercase letter, a digit, or the underscore character.

User response: Correct the name and retry the operation.

HFMDB928

Insufficient authority (short). You cannot revoke this privilege. The current SQLID (*D2 authid*)does not have SYSADM or SYSCTRL authority. You can only revoke a privilege when the current SQLID (*Db2 authid*) originally granted the privilege. (long).

Explanation: The ZDT/Db2 privileges utility was used to attempt to issue a REVOKE statement. The user current Db2® authid *D2 authid*, is different to the Db2® authid originally used to grant the privilege. While it is possible to revoke a privilege originally granted by another Db2® authid, this requires SYSADM or SYSCTRL authority, which the current *Db2 authid* does not have. The statement cannot be executed.

User response: In general, without either SYSADM or SYSCTRL authority, you will be unable to revoke Db2® privileges granted by another Db2® authid. Consult with your Db2® Security Administrator to obtain the necessary Db2® authority to execute the statement.

HFMDB939

LOB limit exceeded (short). A Db2® internal limit for the size of a LOB has been exceeded. Look up the SQLCODE-904 issued previously and respond as appropriate.

Explanation: Db2® reported an SQLCODE-904 error when an attempt was made to add data to a LOB. This error is specific to the XML editor. The current editor session is terminated and the current unit of work is rolled back. Any changes made since the last explicit SAVE are lost.

User response: An SQL error pop-up panel is displayed before this message is issued. Look up the reason code in the message and respond as appropriate. You might need to increase the Db2® installation value LOBVALA, specified in DSNTIJUZ.

HFMDB940

Invalid PIECESIZE (short). PIECESIZE in *Klobytes|Megabytes|Gigabytes* must be a multiple of 2 between *mn value* and *mx value*. (long).

Explanation: An invalid value was entered for "piecesize" on the "Create Index: Type" panel (Db2® Object Functions). *mn value* and *mx value* show the minimum and maximum values that can be used for the specified space-multiplier, which will be either kilobytes, megabytes, or gigabytes.

User response: Select an integer value between the two values shown that is a power of 2. For kilobytes, valid values are: 256, 512, 1024, .. 67,108,864. For Megabytes valid values are: 1, 2, 4 .. 65,536. For Gigabytes, valid values are: 1, 2, 4 .. 64.

HFMDB941

Invalid combination (short) A VCAT name cannot be specified with a storage group name, primary, or secondary space requirements, or the ERASE option.

Explanation: An invalid combination of values was entered when attempting to create a table space, or index, using the Db2® Object Functions. When creating either a table space, or index, a VCAT name can be specified, indicating that space allocation is under user control. Otherwise, the VCAT name must be blank, and a storage group name can be specified, indicating that space allocation is controlled by Db2®. In the latter case, primary and secondary space values may also be entered.

User response: Decide whether a VCAT name or storage group name is appropriate. In the former case do not specify any of: storage group name, primary or secondary space allocation values. In the latter case do not specify a VCAT name value.

HFMDB942

Storage group required (short) Primary and secondary space requirements, and the ERASE option can only be specified if a storage group name is also specified. (long)

Explanation: An invalid combination of values was entered when attempting to create an index using the Db2® Object Functions. Either:

1. One or more non-blank values have been specified for "Primary space" or "Secondary space" on the "Create Index:Allocation" panel. However the "Storage group name" value is blank. You must enter a "storage group name" if you intend to specify space allocation values.
2. One or more non-blank values have been specified for "Primary space", "Secondary space", or "Erase data" on the "Create Index: Partitions" panel. However, the "Storage group name" value is blank. You must enter a "storage group name" if you intend to specify space allocation or an "Erase data" value."

User response: Enter a valid storage group name. Alternatively do not specify "Primary space" "Secondary space" or "Erase data" values.

HFMDB943

Invalid selection (short) Select or deselect columns using the character "S". Order columns explicitly in the *key|index*using numerics.

Explanation: An invalid character was entered in either the Create Table, or Create Index functions, of the Db2® Objects utility.

User response: You can select a column by typing an 'S' in the appropriate field. All other letters are invalid. If a column has previously been selected, you deselect it by typing 'S' in the appropriate field. Once a column has been selected, you can enter any numeric (1, 2, 3, and so on) to specify the order of the columns in the index, or key. On the "Create Table: Unique Constraints " panel you can also select a column by typing any numeric in the appropriate field, and deselect a column by removing the number in the appropriate field.

HFMDB944

Already used (short). Order number *number* was already set for column *column name* on row *row number* of the display.

Explanation: An invalid number was entered in either the "Create Table" or "Create Index" functions of the Db2® Objects utility. When defining an index, or key with multiple columns, the order of the columns in the index, or key, is specified using numbers. Thus, for an index comprising three columns (COL4, COL7 and COL10), you would enter the numbers 1, 2 and 3 (in some combination) against those three columns.

User response: The number for the column has already been used for another column. Determine which columns should be in the index or key, then assign a unique number to each column. The column with the lowest used number will be the first column in the index or key, followed by the column with next-lowest-used number, and so on.

HFMDB945

Maximum partitions (short). The insert (I) or repeat (R) command cannot be executed because the maximum number of partitions allowed by Db2® has already been defined. (long).

Explanation: The user issued the 'I' or 'R' command against the 254th row on the "Create Index: Partitions" panel of the Db2® Objects utility. The maximum number of partitions that can be defined is 254 (Db2® version 7), so extra partitions cannot be added.

User response: No action required.

HFMDB946

Invalid value (short). Enter one of the listed values: *list of valid values*. (long).

Explanation: An invalid value was entered for a field in the Db2® Objects function. *list of valid values* contains the acceptable values for the field in error.

User response: Specify a value shown in *list of valid values*.

HFMDB947

Set columns/partitions (short). You must use option 2 to select the columns comprising the index and option 5 to define the partitions before assigning partition values. (long).

Explanation: The user selected option 6 on the "Create Index" panel of the Db2® Objects function. Either the columns that should comprise the index, or the partitions for the index, have not been defined. Both tasks must be completed before partition values may be assigned.

User response: Select option 2 to confirm that the columns that should comprise the index have been identified; then select option 5 to define the partitions for the new index.

HFMDB948

IT:Define &$ITEM:TE &$ITEM---17 T:You must use option &$OPTN to define the &$ITEM for the &$OBJ :TE T:before using this option.:TE

Explanation: T

User response: I

HFMDB949

Table not found (short). The table to be indexed was not found in the Db2® catalog. (long).

Explanation: The user entered a table name on the "Create Index" panel of the Db2® Objects function. That table name does not exist on the current server.

User response: Check the table name for spelling mistakes. You can use the ZDT/Db2 Object List utility to show all table names defined at the current Db2® server if required. Correct the table name and retry the operation.

HFMDB950

Set partition values (short) Partitions have been defined for this index using option 5. You must set partition values using option 6 before you can define the index.

Explanation: This message is issued by the "Create Index" dialog (Db2® Object Functions). For a partitioned index, the number of partitions is set using option 5 (Partitions). Once the partitions are defined, you also need to set the maximum values for each partition using option 6 (Partition values). The index cannot be created until the partition values have been specified.

User response: If a partitioned index is required, select option 6 (Partition values), and specified appropriate values. If a non-partitioned index is required, select option 5 (Partitions), and delete all but one partition entry.

HFMDB951

Update partition values (short) The column selections or partition definitions have changed since the partition values were set. You must reset the partition values using option 6 before you can define the index.

Explanation: This message is issued by the "Create Index" dialog (Db2® Object Functions). Partition values have been set for all partitions and columns previously specified. However, a subsequent change has invalidated the partition values. There are two possibilities:

1. The columns specified for the new index have changed (columns added or deleted).
2. The partitions specified using option 5 (Partitions) have changed. The index cannot be created until the partition values have be specified.

User response: If a partitioned index is required, select option 6 (Partition values) and specified appropriate values. If a non-partitioned index is required, select option 5 (Partitions) and delete all but one partition entry.

HFMDB952

Conflicting selection (short) *value 1* > is not valid because *value 2*.

Explanation: This message is issued in many places within the Db2® Object Function dialog. The SQL syntax used to create the object has restrictions on the use of both *value 1* and *value 2*. In most cases you can resolve the error by specifying either *value 1* or *value 2*, but not both. The Object cannot be created until the conflicting selections are resolved.

User response: Determine which of the two selections is appropriate and remove the unwanted selection. Retry the operation. If the appropriate choice cannot be determined, refer to the SQL Reference manual for the appropriate version of Db2®.

HFMDB953

Errors retained (short) Invalid data has been saved because the CANCEL command was entered in response to an input verification message. The *statement type* statement may be invalid if executed: return to the previous panel and correct the error, or return to the *panel name* panel and start from the beginning.

Explanation: This message is issued in many places within the Db2® Object Function dialog. An invalid value was entered on an option panel for the current CREATE or DROP function, followed by the user pressing CANCEL (PF12 by default) to return to the option selection panel for the current CREATE or DROP function. The invalid value is unresolved and may cause problems if an attempt is made to executed the current CREATE or DROP function.

User response: In most cases the problem can be resolved by reselecting the previous option panel (where CANCEL was entered), and correcting the invalid values entered there. If this does not resolve the problem, return to the Db2® Object Functions panel and restart the operation.

HFMDB954

No suitable rows (short). The current table definition does not contain columns specified as not null. Use option 2 to define not null columns before defining :primary/unique keys. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions). The user selected the "unique constraints" option after previously defining columns for the new table using option 1 (Columns). A column can only be included in a unique constraint key if it is defined as NOT NULL, and none of the columns currently defined for the new table meet this requirement.

User response: Select option 2 (Nulls/Default values) and set the Null status for each column as appropriate. You will need to specify either NN (NOT NULL) or ND (NOT NULL WITH DEFAULT) for at least one column to be able to specify a unique constraint for the new table.

HFMDB955

No suitable rows (short). The current table definition does not contain any non-ROWID, non-LOB columns specified as not null, nor any identity columns. Use option 2 to define not null columns, or option 8 to define identity columns before defining primary/unique keys. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions). The user selected the "unique constraints" option after previously defining columns for the new table using option 1 (Columns). A column can only be included in a unique constraint key if it is defined as NOT NULL, and none of the columns currently defined for the new table meet this requirement. This message is issued when there are columns defined as ROWID, or CLOB/DBCLOB, BLOB.

User response: Select option 2 (Nulls/Default values), and set the Null status for each column as appropriate. You will need to specify either NN (NOT NULL) or ND (NOT NULL WITH DEFAULT) for at least one column to be able to specify a unique

constraint for the new table. You can also define numeric columns as INDENTITY columns using option 8 (Generate values). An IDENTITY column can be included in a unique constraint.

HFMDB956

Too many selections (short). You have already selected *Max column count* columns for this key, which is the maximum allowed by Db2®. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions), usually when defining columns to be included in the primary key, or other unique constraint defined for the table. *Max column count* is 64 for Db2® for z/OS® versions 8 - 10 inclusive.

User response: Reduce the number of columns in the primary key or unique constraint.

HFMDB957

Table/view not found (short). The model table/view was not found in the Db2® catalog. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions), when a model table name is entered, but the Db2® object does not exist.

User response: Correct the Db2® object name, retry the operation. You can use the Object List utility (Option 3.4) to show all the Db2® objects (tables/views/aliases) defined at the local server.

HFMDB958

Too many constraints (short) Too many unique constraints have been found in model tables. Only the first three unique constraints found in the Db2® catalog have been loaded.

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions), when a model table name is entered and a load of the informationm for the model table is requested. The "Create table" dialog supports a maximum of three unique constraints for any table being defined. This message might be issued for the load of a single Db2® object, with 4 or more unique constraints defined, or when multiple loads are requested.

User response: This is an informational message. In the unlikely event that 4 or more unique constraints need to be defined for the table being created, consider defining the table with 3 initial constraints, and using ALTER TABLE to add any additional constraints.

HFMDB959

Invalid option (short). Only column information can be loaded when the add option is selected and columns have already been defined or loaded. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions), when a model table name is entered. The user loaded information for one model table, then attempted to add load additional information for another. For the second and subsequent loads, only the "Basic table information" and "Column information" options can be selected.

User response: You can only load "Primary key information", "Unique key information", "Referential constraints" and "Check constraint" information for the first model table specified. Deselect the option for all of the above and retry the operation. Alternatively set the Refresh or Add option to Refresh and repeat the operation. This will replace any existing information with that for the current model table.

HFMDB960

Too many constraints (short) Too many table foreign keys have been defined for this table. The number of table foreign keys is limited to the number of ISPF tables named HFM2CTnn that can be created.

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions). An attempt was made to create more than 100 foreign keys for the current table. While there is no Db2® limit on the number of foreign keys, the "Create Table" dialog is limited to a maximum of 100 foreign keys.

User response: Review the number of foreign keys that have been specified, reduce the number to 100 or less.

HFMDB961

Order required (short) A referenced column name has been entered for this column, but no order. An order is required if the column is to be included in the foreign key.

Explanation: This message is issued on the "Create Table: Table Referential Constraint" panel. When creating a table referential constraint that includes more than one column (the usual usage is for a multiple-column foreign key definition), an order specification is required for every column that is part of the foreign key being defined.

User response: Specify an order for every column that is to be part of the foreign key.

HFMDB962

Ref. column required (short). If any of the foreign key columns reference a specific column in the parent table, then all of the foreign key columns must do so. (long).

Explanation: This message is issued on the "Create Table: Table Referential Constraint" panel. When creating a table referential constraint that includes more than one column (the usual usage is for a multiple-column foreign key definition), an order specification for a column was entered without a corresponding entry in the Parent Key column field.

User response: Specify a column name in the Parent Key column field, for every column that is to be part of the foreign key.

HFMDB963

No columns selected (short). No columns have been selected to define the foreign key for this constraint. (long).

Explanation: This message is issued when creating referential constraints for a table (Create Table dialog of Db2® Object functions). The user created a new constraint name and specified the parent table for the relationship. However no foreign key columns have been defined.

User response: Use the 's' command to select the new constraint entry. When the "Create Table: Table Referential Constraint" panel is displayed, select the columns that comprise the foreign key, and enter the names of the corresponding columns in the parent table.

HFMDB964

Some data not loaded (short). The Db2® catalog did not contain any information for the model table or view for at least one of the model load options specified. For example, check constraints were requested but the referenced model table has no check constraints. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions), when a model table name is entered, and a load of existing Db2® object information is requested.

User response: No action required. Informational message.

HFMDB965

No suitable rows (short) The current table definition does not contain any short string columns specified as not null. Use option 2 to define not null columns before defining Procedure Exits.

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions). The user selected option 7 (Procedure exits). However, there are no suitable columns defined on which a procedure exit can be defined. A procedure exit cannot be defined unless a suitable column is defined.

User response: Option 7 (Procedure exits) allows a FIELDPROC specification for suitable columns of the table. In general, only character (CHARACTER/GRAPHIC) columns of length <=255, defined as NOT NULL, or NOT NULL WITH DEFAULT, are suitable. Define a suitable column using options 1 (Columns) and 2 (Nulls/Default values) and then reselect option 7.

HFMDB966

External name required (short). Specify an external name for the *object type* using option *number* before using this option. This is required because the specified name exceeds eight characters. (long).

Explanation: This message is issued by the "Create Function" and "Create Stored Procedure" dialogs of the Db2® Object Functions. The length of the name for the function, or stored procedure, exceeds 8 characters. In this situation an external name for the function must also be specified. *object type* is either "function" or "stored procedure", depending on the Object function. *number* is 4 when creating a function, and 3 when creating a stored procedure.

User response: Select the specified option, and enter an external name for the function or stored procedure. Consult the Db2® for z/OS® SQL Reference for the appropriate version of Db2® for details of acceptable formats for the external name value.

HFMDB967

String too long (short). The string constant specified as a default value is too long for the associated string column. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when specifying a default value for a character data type column. The default value is too long. For example, specifying a default value of 'ABCDE' (length 5) for a CHAR(3) (length 3) column.

User response: Compare the length of the default value string, and the maximum length for the column, as defined in the length field for the data type. Adjust either value as appropriate.

HFMDB968

Invalid data (short). The decimal number specified as a default value is invalid or out of range for the associated decimal number column. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when specifying a default value for a DECIMAL data type column. The specified default value is invalid in the context of the precision and scale specified for the column.

User response: Compare the definition for the DECIMAL column with the implied precision, and scale from the default value. Adjust either the data type (precision and scale) in the column, or the default value.

HFMDB969

Invalid *DATE*|*TIME*|*TIMESTAMP* (short) The format of the *DATE*|*TIME*|*TIMESTAMP* field or the data it contains is invalid.

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when specifying a default value for a DATE, TIME, or TIMESTAMP data type column. The specified default value is invalid.

User response: Examine the default value entered and the data type for the column. The two possible causes of this error are:

1. An invalid format. Example: '12/10/2001' is not a valid format for a TIME column.
2. An invalid value. Example: '32/11/2001' is an invalid date, since 32 is not a valid day (or month).

Consult the Db2® for z/OS® SQL Reference for the appropriate version of Db2® for a detailed description of valid DATE/TIME and TIMESTAMP constants, including supported formats.

HFMDB970

Date/time not validated (short). The default values for one or more date or time fields were not validated because the Db2® subsystem date/time format is LOCAL or unknown.

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when specifying a default value for a DATE or TIME column. The Db2® default for either the DATE or TIME format is either LOCAL, or unknown. In either case, the panel logic cannot validate the specified default value.

User response: No action required. Informational message. If any default value entered is invalid, an SQL error will occur when the attempt is made to create the Db2® object.

HFMDB971

Date/time not validated (short) The specified default date or time value was not validated because the Db2® subsystem date/time format is LOCAL or unknown. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when specifying a default value for a DATE, TIME, or TIMESTAMP column. This message is issued on pop-up panel "Create Table: Default Value". The Db2® default for either the DATE or TIME format is either LOCAL, or unknown. In either case, the panel logic cannot validate the specified default value.

User response: No action required. Informational message. If any default value entered is invalid an SQL error will occur when the attempt is made to create the Db2® object.

HFMDB972

Invalid data (short). The floating point number specified as a default value is invalid or outside the range supported by Db2®. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when specifying a default value for a binary floating point column (REAL, FLOAT, DOUBLE). The specified default value is outside the range supported by Db2®.

User response: The limits for floating point constants are documented in "Limits in Db2® for z/OS®", in the Db2® for z/OS® SQL Reference manual, for the appropriate version of Db2®. Compare the specified default value against the documented values, and adjust as appropriate.

HFMDB973

GENERATED ALWAYS added (short). The GENERATED ALWAYS clause has been added to the definition of the column with a data type of ROWID. To change the GENERATED clause select option 8. (long).

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions), when adding a column definition for a ROWID column.

User response: No action required. Informational message. In most cases it should not be necessary to modify the GENERATED clause for a ROWID column. Option 8 (Generated Values) allows the GENERATED clause to be modified.

HFMDB974

Invalid database name (short). You cannot specify the name of an implicitly created database when creating a table. Implicitly created database names have the form DSNxxxxx, where xxxxx is any 5 numeric digits. (long)

Explanation: This message is issued by the "Create Table" dialog (Db2® Object Functions) when attempting to create a table. The database name specified is invalid because it has the form 'DSNxxxxx', where xxxxx is 00000-99999 inclusive. Names of this form are reserved for use by Db2®, as implicitly created database names. The table cannot be created.

User response: Specify a different Db2® database name, or remove the database name completely if an implicitly required database is acceptable.

HFMDB980

Invalid value (A/D) (short). The valid values for the A/D field are A, for ascending; D, for descending, or blank. (long).

Explanation: This message is issued on the "Column Selection/Edit" panel, when a value other than 'A' or 'D' is entered in the "A/D" column.

User response: The "A/D" column in the "Column Selection/Edit" panel is used to specify the sort order for col-umns included in the ORDER BY clause. The valid values are 'A' for ascending (the default) or 'D' for descending. Specify one of these values, or leave the field blank for the default behaviour.

HFMDB981

Enter sort order (short). A sort order is required when a sort column is specified (A or D in the A/D field). Enter a number in the range 1..999. (long).

Explanation: The 'Order' column in the "Column Selection/Edit" panel is used to specify the sort sequence for columns included in the ORDER BY clause. The user specified either 'A' or 'D' in the "A/D" column, but no value in the 'Order' column.

User response: Enter a number in the 'Order' column for every column that also has a non-blank value in the "A/D" column.

HFMDB990

Mutually exclusive value (short). Specify a JOBNAME value, or an ADDRESS value, but not both. (long).

Explanation: This message is issued by the "Create Trusted Context" dialog (Db2® Object Functions). A value was entered for both the JOBNAME and ADDRESS fields, however only one value is acceptable.

User response: Specify either a JOBNAME value; or an ADDRESS value, but not both. The syntax for the CREATE TRUSTED CONTEXT SQL statement can be found in the Db2® for z/OS® SQL Reference manual for the appropriate version of Db2®.

HFMDB991

Value required (short). Either a JOBNAME value or an ADDRESS value is required. (long).

Explanation: This message is issued by the "Create Trusted Context" dialog (Db2® Object Functions). No value was entered in either the JOBNAME or ADDRESS fields; a value is required for one.

User response: Specify either a JOBNAME value; or an ADDRESS value, but not both. The syntax for the CREATE TRUSTED CONTEXT SQL statement can be found in the Db2® for z/OS® SQL Reference manual for the appropriate version of Db2®.

HFMDB992

Value not required (short). Do not specify a "Default Role Name" value when the "No Default Role" option is selected. (long).

Explanation: This message is issued by the "Create Trusted Context" dialog (Db2® Object Functions). The user selected the "No default role" option, and also entered a value in the "Default role name" field. The two selections are mutually exclusive.

User response: If a default role name is to be specified, deselect the "No default role" option. If there is to be no default role name, remove the "Default role name".

HFMDB993

Value not required (short). Do not specify a "Security Label Name" value when the "No Default Security Label" option" is selected. (long).

Explanation: This message is issued by the "Create Trusted Context" dialog (Db2® Object Functions). The user selected the "No default security label" option, and also entered a value in the "Security label name" field. The two selections are mutually exclusive.

User response: If a default security label name is to be specified, deselect the "No default security label" option. If there is to be no default security label, remove the "Security label name".

HFMDB994

Value not required (short). Do not select the "Without authentication" option when the "PUBLIC" option" is selected. (long).

Explanation: This message is issued by the "Create Trusted Context" dialog (Db2® Object Functions). The user did not select the "PUBLIC" use option, and selected the "Without authentication" option. The second selection requires that the first option also be selected.

User response: The valid choices for the two options ("PUBLIC" and "Without authentication") are: 1). Both options not selected. 2) Both options selected. 3) Only the first option selected.

HFMDC014

Rows rows exported (short). *Rows* row(s) exported from SELECT statement (user-specified) to *file data set name*. (long).

Explanation: This is a message insert used by the ZDT/Db2 Export function. It shows the number of rows rows exported by the export function. The target data set is *data set name*. This message is specific to the export of data from a user-specified SELECT statement initially entered using ZDT/Db2 options 4.1-4.4.

User response: No action required. Informational message.

HFMDC015

Rows rows+hdr expd (short). *Rows* row(s) and column header row exported from SELECT statement (user-specified) to *file data set name*. (long).

Explanation: This is a message insert used by the ZDT/Db2 Export function. It shows the number of rows *rows* exported by the export function. The target data set is *data set name*. This message is specific to the export of data in CSV format, with column header data, from a user-specified SELECT statement initially entered using ZDT/Db2 options 4.1-4.4.

User response: No action required. Informational message.

HFMDC019

Not in large mode (short). The export command with the ALL|X|EX|NX parameter is not available in large editor mode (row count = 0). (long).

Explanation: The editor EXPORT command with the ALL|X|EX|NX parameter specified was issued from a Z Data Tools/Db2 editor session, in large mode (row count = 0, also ALL, *). This is not supported and the command cannot be processed.

User response: Use normal editor mode and re-try the operation.

HFMDC020

count row(s) exported from Db2® object *source Db2 object* to file *file name*. The open returned SQLCODE *sqlcode*. *trunc_count* rows had fetch SQLCODE +445 (truncation) warnings. *warn_count* other rows had fetch warnings (SQLCODE>0, but not +445).

Explanation: This message is issued at the end of a Z Data Tools/Db2 export function, when at least one positive SQLCODE was received retrieving data from Db2®. The number of rows exported, *count*; the Db2® object from which the rows were exported, *source Db2 object*; the output file name that contains the exported data, *file name*; the SQLCODE for the OPEN cursor, *sqlcode*; the number of rows fetched that had SQLCODE+445 warnings, *trunc_count* and the number of rows fetched that gave positive SQLCODEs (but not +445), *warn_count* are shown. SQLCODE+445 is a warning condition issued when data truncation occurs.

User response: For SQLCODE+445 warnings, examine the exported data to see if the data truncation has resulted in any unexpected consequences. Take action as appropriate. For other warning SQLCODEs: If the export operation was run on-line, repeat the operation using the batch execution option. Run the resulting job and examine the messages produced. Look up the SQLCODEs reported in the Db2® for z/OS® SQL Codes manual, for the appropriate version of Db2®. Take action as appropriate.

HFMDC021

Row count row(s) and column header row exported from Db2® object *object name* to file *data set name*. The open returned SQLCODE *sqlcode*. *trunc_count* rows had fetch SQLCODE +445 (truncation) warnings. *warn_count* other rows had fetch warnings (SQLCODE>0, but not +445).

Explanation: This message is issued by the ZDT/Db2 export function, when at least one positive SQLCODE was received retrieving data from Db2® and is specific for an export with CSV format and the "Include column headers" option selected. The number of rows exported, *count*; the Db2® object from which the rows were exported, *source Db2 object*; the output file name that contains the exported data, *file name*; the SQLCODE for the OPEN cursor, *sqlcode*; the number of rows fetched that had SQLCODE+445 warnings, *trunc_count* and the number of rows fetched that gave positive SQLCODEs (but not +445), *warn_count* are shown. SQLCODE+445 is a warning condition issued when data truncation occurs.

User response: For SQLCODE+445 warnings, examine the exported data to see if the data truncation has resulted in any unexpected consequences. Take action as appropriate. For other warning SQLCODEs: If the export operation was run on-line, repeat the operation using the batch execution option. Run the resulting job and examine the messages produced. Look up the SQLCODEs reported in the Db2® for z/OS® SQL Codes manual, for the appropriate version of Db2®. Take action as appropriate.

HFMDC022

Rows row(s) exported from SELECT statement (user-specified) to file *data set name*. The open returned SQLCODE *sqlcode*. *trunc_count rows* had fetch SQLCODE +445 (truncation) warnings. *warn_count* other rows had fetch warnings (SQLCODE>0, but not +445).

Explanation: This message is issued by the ZDT/Db2 Export function when at least one positive SQLCODE was received retrieving data from Db2®. It shows the number of rows *rows* exported by the export function; the target data set is *data set name*; the SQLCODE for the OPEN cursor is *sqlcode*; the number of rows fetched that had SQLCODE+445 warnings is *trunc_count* and the number of rows fetched that gave positive SQLCODEs (but not +445) is *warn_count*. SQLCODE+445 is a warning condition issued when data truncation occurs. This message is specific to the export of data from a user-specified SELECT statement initially entered using ZDT/Db2 options 4.1-4.4.

User response: For SQLCODE+445 warnings, examine the exported data to see if the data truncation has resulted in any unexpected consequences. Take action as appropriate. For other warning SQLCODEs: If the export operation was run on-line, repeat the operation using the batch execution option. Run the resulting job and examine the messages produced. Look up the SQLCODEs reported in the Db2® for z/OS® SQL Codes manual, for the appropriate version of Db2®. Take action as appropriate.

HFMDC023

Rows row(s) and column header row exported from SELECT statement (user-specified) to file *data set name*. The open returned SQLCODE *sqlcode*. *trunc_count* rows had fetch SQLCODE +445 (truncation) warnings. *warn_count* other rows had fetch warnings (SQLCODE>0, but not +445).

Explanation: This message is issued by the ZDT/Db2 Export function when at least one positive SQLCODE was received retrieving data from Db2®. It shows the number of rows *rows* exported by the export function; the target data set is *data set name*; the SQLCODE for the OPEN cursor is *sqlcode*; the number of rows fetched that had SQLCODE+445 warnings is *trunc_count* and the number of rows fetched that gave positive SQLCODEs (but not +445) is *warn_count*. SQLCODE+445 is a warning condition issued when data truncation occurs. This message is specific to the export of data in CSV format, with column header data, from a user-specified SELECT statement initially entered using ZDT/Db2 options 4.1-4.4.

User response: For SQLCODE+445 warnings, examine the exported data to see if the data truncation has resulted in any unexpected consequences. Take action as appropriate. For other warning SQLCODEs: If the export operation was run on-line, repeat the operation using the batch execution option. Run the resulting job and examine the messages produced. Look up the SQLCODEs reported in the Db2® for z/OS® SQL Codes manual, for the appropriate version of Db2®. Take action as appropriate.

HFMDC026

Open cursor SQLCODE=*sqlcode*.

Explanation: This message is issued by the ZDT/Db2 export function, when executed in batch mode, when a positive SQLCODE is received from Db2® when the cursor is opened.

User response: Look up the SQLCODE reported in the Db2® for z/OS® SQL Codes manual, for the appropriate version of Db2®. Take action as appropriate.

HFMDC027

Fetch SQLCODE=*sqlcode* for row *row* (data truncation).

Explanation: This message is issued by the ZDT/Db2 export function, when executed in batch mode, whenever an SQLCODE +445 is received when fetching a row of data from Db2®. SQLCODE+445 indicates that data truncation occurred.

User response: Examine the exported data to see if the data truncation has resulted in any unexpected consequences. Take action as appropriate.

HFMDC028

Fetch SQLCODE=*sqlcode* for row *row* (warning).

Explanation: This message is issued by the ZDT/Db2 export function, when executed in batch mode, whenever a positive SQLCODE other than +445 is received when fetching a row of data from Db2®.

User response: Look up the SQLCODE reported in the Db2® for z/OS® SQL Codes manual, for the appropriate version of Db2®. Take action as appropriate.

HFMDC030

Processing Db2® object name for DELETE.

Explanation: This informational message is issued at the start of a BATSQL function. ZDT/Db2 is processing object name, for delete processing.

User response: No action is required.

HFMDC031

Inserting rows into Db2® object name.

Explanation: This informational message is issued at the start of a BATSQL function. ZDT/Db2 is processing object name, for insert processing.

User response: No action is required.

HFMDC032

Processing Db2® object name for UPDATE.

Explanation: This informational message is issued at the start of a BATSQL function. ZDT/Db2 is processing object name, for update processing.

User response: No action is required.

HFMDC033

Row count rows deleted.

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 successfully deleted row count rows from the object.

User response: No action is required.

HFMDC034

Row count rows inserted.

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 successfully inserted row count rows into the target object.

User response: No action is required.

HFMDC035

Row count rows updated.

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 successfully updated row count rows from the object.

User response: No action is required.

HFMDC036

Batch DELETE canceled because of excessive errors (*errcnt*,*maxerr*).

Explanation: This informational message is issued during TYPE=DELETE BATSQL processing. ZDT/Db2 stopped deleting rows because the user-specified error limit *maxerr* (MAXERRORS keyword) was reached. The actual number of errors was *errcnt*. A summary of the errors that are encountered follows (messages HFMDC039 and HFMDC060). Note that when an error is encountered processing a rowset, the error count is incremented by the number of rows in the rowset. This can result in the actual number of errors reported exceeding the value specified for MAXERRORS.

User response: Examine the error summary and respond.

HFMDC037

Batch INSERT canceled because of excessive errors (*errcnt*,*maxerr*).

Explanation: This informational message is issued during TYPE=INSERT BATSQL processing. ZDT/Db2 stopped inserting rows because the user-specified error limit *maxerr* (MAXERRORS keyword) was reached. The actual number of errors was *errcnt*. A summary of the errors that are encountered follows (messages HFMDC039 and HFMDC060).

User response: Examine the error summary and respond.

HFMDC038

Batch UPDATE canceled because of excessive errors (*errcnt*,*maxerr*).

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 stopped updating rows because the user-specified error limit (MAXERRORS keyword) was reached. A summary of the errors that are encountered follows (message HFMDC039). Note that when an error is encountered processing a rowset, the error count is incremented by the number of rows in the rowset. This can result in the actual number of errors reported exceeding the value specified for MAXERRORS.

User response: Examine the error summary and respond.

HFMDC039

Error count rows gave SQLCODE sqlcode.

Explanation: This informational message is issued at the end of a BATSQL function. During the function, the delete/insert or update operation resulted in error count SQLCODE sqlcode errors. If there were different SQLCODEs received (for example SQLCODE-803, SQLCODE-545) there is one occurrence of this message for each different SQLCODE.

User response: Examine the error summary and respond.

HFMDC040

Row count pending deletes rolled back.

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 stopped deleting rows because the user-specified error limit (MAXERRORS keyword) was reached. Row count rows were deleted after the last commit point. The changes to these rows were backed out as a result of the Db2® rollback.

User response: No action is required.

HFMDC041

Row count pending inserts rolled back.

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 stopped inserting rows because the user-specified error limit (MAXERRORS keyword) was reached. Row count rows were inserted after the last commit point. The changes to these rows were backed out as a result of the Db2® rollback.

User response: No action is required.

HFMDC042

Row count pending updates rolled back.

Explanation: This informational message is issued at the end of a BATSQL function. ZDT/Db2 stopped updating rows because the user-specified error limit (MAXERRORS keyword) was reached. Row count rows were updated after the last commit point. The changes to these rows were backed out as a result of the Db2® rollback.

User response: No action is required.

HFMDC043

Row count commit(s) issued prior to completion.

Explanation: This informational message is issued at the end of a BATSQL function. The AUTOCOMMIT keyword specified a nonzero value. ZDT/Db2 issued one or more commits as rows were processed. Row count does not include the final commit or rollback issued when the function ends.

User response: No action is required.

HFMDC044

Required SET clause missing.

Explanation: This error message is issued during processing of a BATSQL function. An update operation was requested, but no SET clause was found.

User response: Ensure that a SET clause is specified, rerun the function.

HFMDC045

SET clause must begin with 'SET'

Explanation: This error message is issued during processing of a BATSQL function. An update operation was requested, however the SET clause does not commence with 'SET'.

User response: The SET clause must be a valid SET clause for an update statement, see the Db2® SQL Reference manual for a full description. Ensure that the SET clause begins with 'SET', rerun the function.

HFMDC046

SET clause incomplete, it must contain an assignment for >= 1 column(s).

Explanation: This error message is issued during processing of a BATSQL function. An update operation was requested, however the SET clause does not include an assignment for at least 1 column.

User response: The SET clause must include an assignment for at lease one column, for example: "SET COL1=value". See the Db2® SQL Reference manual for a full description. Correct the SET clause, rerun the function.

HFMDC047

WHERE clause must begin with 'WHERE'.

Explanation: This error message is issued during processing of a BATSQL function. An update or delete operation was requested, however the WHERE clause does not commence with 'WHERE'.

User response: The WHERE clause must be a valid WHERE clause for an update or delete statement, see the Db2® SQL Reference manual for a full description. Ensure that the WHERE clause begins with 'WHERE', rerun the function.

HFMDC048

Isolation UR incompatible with delete or update, CS substituted.

Explanation: This informational message is issued at the start of a BATSQL function. The control statements include ISOLATION=UR for either a delete or update function. Uncommitted read is not supported when a positioned delete or update statement is issued. This method is the method that is used by the BATSQL function for deleting or updating rows. The isolation value was to CS (Cursor stability).

User response: Review the value that is specified for the ISOLATION keyword. Consider changing the value to 'NONE' or 'CS'.

HFMDC049

DELETEALL=YES must be specified for a delete without a WHERE clause.

Explanation: This error message is issued during processing of a delete BATSQL function. For a delete function, the omission of a where clause results in the attempted deletion of all rows of the table. As a safety feature, DELETEALL=YES must also be included in the control statements.

User response: Review the intended action of the BATSQL function. If the intention is to delete every row in the table, add the DELETEALL=YES statement and rerun the function.

HFMDC050

Insert column list must be enclosed in (). Eg (Col1, Col2).

Explanation: This error message is issued during processing of an insert BATSQL function. An invalid insert column clause was detected.

User response: The insert column clause must include a list of comma-separated column names. The entire list requires enclosing parentheses. See the Db2® SQL Reference manual for a full description. Correct the insert column clause, rerun the function.

HFMDC051

Insert column list incomplete, it must contain >=1 column(s).

Explanation: This error message is issued during processing of an insert BATSQL function. An invalid insert column clause was detected.

User response: The insert column clause must contain at least one column name. Further, that column name must be a column of the target table. See the Db2® SQL Reference manual for a full description. Correct the insert column clause, rerun the function.

HFMDC052

Single row processing active.

Explanation: This informational message is issued at the start of a BATSQL function. Single-row processing is active, that is, ZDT/Db2 fetches rows one-at-a-time from the source table.

User response: No action is required.

HFMDC053

Rowset processing (*nnn* rows) active.

Explanation: This informational message is issued at the start of a BATSQL function. Rowset processing is active, that is, ZDT/Db2 fetches *nnn* rows at a time.

User response: No action is required.

HFMDC054

Required SELECT clause is missing or incomplete.

Explanation: This is an error message issued during processing of the BATSQL function. The specified SELECT clause has either not been specified, or is incomplete.

User response: Correct the coding of the SELECT clause. The SELECT clause should start with 'SELECT' and identify a result table. The shortest valid SELECT clause is 'SELECT * FROM <name>', where <name> is an unqualified Db2® object name. Retry the BATSQL function.

HFMDC055

SELECT clause must begin with 'SELECT'.

Explanation: This is an error message issued during processing of the BATSQL function. The specified SELECT clause does not start with 'SELECT'.

User response: Correct the coding of the SELECT clause. Retry the BATSQL function.

HFMDC056

SELECT clause is invalid, SQLCODE=SQLCODE.

Explanation: This is an error message issued during processing of the BATSQL function. The specified SELECT clause is invalid, Db2® returned SQLCODE=SQLCODE in response to an attempt to prepare the statement.

User response: Look up the SQLCODE in the Db2® Codes manual for the appropriate version of Db2® and use the information provided to correct the SELECT statement. Retry the BATSQL function.

HFMDC057

Column COLNAME is not a column of OBJNAME.

Explanation: This is an error message issued during processing of the BATSQL function. The specified column name COLNAME does not exist in Db2® object OBJNAME.

User response: Determine the reason why the specified column name is not a column of the named Db2® object. A non-exhaustive list of causes is:

1. Spelling errors
2. Upper/lower case issues
3. References to an incorrect Db2® object name
4. Referring to a Db2® object name on a different Db2® sub-system to the intended system.

Correct the input statements and retry the BATSQL function.

HFMDC058

The requested function is not supported: XML columns.

Explanation: This is an error message issued during processing of an INSERT BATSQL function. XML columns were detected in either the target Db2® object, or in the source SELECT statement. The BATSQL function with TYPE=INSERT does not support XML columns. The function cannot be performed.

User response: None, the requested operation cannot be performed.

HFMDC059

Switching to single-row processing because of multi-row errors.

Explanation: This is an error message issued during processing of a DELETE or UPDATE BATSQL function. Rowset processing is active, however an SQL error was encountered when attempting a delete/update of a rowset. All rows in the current rowset are unchanged, ZDT/Db2 marks all rows in the rowset as 'in error'. ZDT/Db2 moves to the next rowset and attempts to process each row in that rowset individually.

User response: Examine the output object to determine why an SQL error occured. Take appropriate action.

HFMDC060

1 rowset gave SQLCODE *sqlcode*.

Explanation: This message is issued during processing of a DELETE or UPDATE BATSQL function when rowset processing is active. An SQL error occured when attempting a rowset delete or update, the failing SQLCODE is *sqlcode*. All rows in the current rowset are unchanged, ZDT/Db2 marks all rows in the rowset as 'in error'.

User response: Examine the output object to determine why an SQL error occured. Take appropriate action.

HFMDC062

Invalid SET clause.

Explanation: This message is issued during processing of an UPDATE BATSQL function. The SET clause is invalid. This message is specific to the following form of a SET clause:

```
SET (Col1 , ...) = (Val1, ...)
```

See the SQL Reference for the appropriate version of Db2® for a full description of the SET clause syntax. This error is usually the result of:

- A missing parenthesis around either the column, or value list.
- A missing '=' between the column and value lists.
- The column list has no columns.
- The value list has no values.

User response: Examine the SET clause, correct the syntax error(s), retry the function.

HFMDC170

Changes discarded (short). The invalid change to the data format option has been discarded. Enter one of the following values: 1, 2, 3, 4. (long).

Explanation: The user made an invalid change to the "Data format" value on the Import options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the import options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC171

Changes discarded (short). The invalid change to the duplicate key processing option has been discarded. Enter one of the following values: 1, 2. (long).

Explanation: The user made an invalid change to the "Duplicate key processing" value on the Import options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the import options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC172

Changes discarded (short). The invalid change to the maximum duplicates option has been discarded. Enter one of the following values: *, ALL, or a number in the range 0-2147483647 inclusive. (long).

Explanation: The user made an invalid change to the "Max duplicates" value on the Import options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the import options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC174

Changes discarded (short). The invalid change to the auto-commit count option has been discarded. Enter a number in the range 0-2147483647 inclusive. (long).

Explanation: The user made an invalid change to the "Auto commit count" value on the Import options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the import options panel is re-displayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC180

Enter a value (short). A value is required. Enter a frequency between 0 and 1 exclusive. As an example 0.1 will sample approximately 10% of rows. (long).

Explanation: The user selected "random sampling" on the ZDT/Db2 Editor options panel, but did not enter a value for "sampling frequency". A sampling frequency is required when using random sampling.

User response: Enter a decimal value between 0.0 and 1.0. As an example, enter 0.1 will result in approximately 10% of the rows in the object being sampled. These rows will appear in the ZDT/Db2 editor session. Higher values will result in more rows being sampled; lower values will result in less rows being sampled.

HFMDC181

Invalid value (short). Valid frequencies are 0.000000001 - 0.999999999 inclusive. Enter a value within these limits. (long).

Explanation: The user selected "random sampling" on the ZDT/Db2 Editor options panel, but did not enter a valid value for "sampling frequency". A valid sampling frequency is a decimal number between 0.0 and 1.0; with 9 digits of precision after the decimal point.

User response: Enter a decimal value between 0.0 and 1.0. As an example, enter 0.1 will result in approximately 10% of the rows in the object being sampled. These rows will appear in the ZDT/Db2 editor session. Higher values will result in more rows being sampled; lower values will result in less rows being sampled.

HFMDC182

Enter a number (short). Enter a number (seed) to start the generation of pseudo-random numbers, or 0 to use a seed derived from the system clock. Valid values are 0-2147483647 inclusive. (long).

Explanation: The user selected "random sampling" on the ZDT/Db2 Editor options panel, but did not enter a valid value for "sampling seed". The seed value is used to initiate the pseudo-random number generator.

User response: Enter an integer between 0-2147483647 inclusive. A value of 0 sets the seed to a value based on the current value of the system clock on the computer, accurate to the microsecond, and is therefore a random value for most purposes. Any other value will result in repeatable behaviour, in terms of the rows sampled, for a given Db2® object and the same sampling frequency.

HFMDC183

Changes discarded (short). The invalid change to the Initial skip count has been discarded. The Initial skip count value must be in the range 0-999999999. (long).

Explanation: The user made an invalid change to the "Initial skip count" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC184

Changes discarded (short). The invalid change to the Include count has been discarded. The Include count value must be in the range 1-999999999. (long).

Explanation: The user made an invalid change to the "Include count" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC185

Changes discarded (short). The invalid change to the Skip count has been discarded. The Skip count value must be in the range 0-999999999. (long).

Explanation: The user made an invalid change to the "Skip count" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC186

Changes discarded (short). The invalid change to the Sampling limit has been discarded. The Sampling limit value must be in the range 0-999999999. (long).

Explanation: The user made an invalid change to the "Sampling limit" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC187

Changes discarded (short). The invalid change to the Sampling frequency has been discarded. The Sampling frequency value must be in the range 0.000000001 - 0.999999999. (long).

Explanation: The user made an invalid change to the "Sampling frequency" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC188

Changes discarded (short). The invalid change to the Sampling seed has been discarded. The Sampling seed value must be in the range 0-2147483647. (long).

Explanation: The user made an invalid change to the "Sampling seed" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC189

Changes discarded (short). The invalid change to the Change count before auto commit has been discarded. The Change count before auto commit value must be in the range 0-2147483647. (long).

Explanation: The user made an invalid change to the "Change count before auto commit" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC190

Changes discarded (short). The invalid change to the Prefix Length option has been discarded. (long).

Explanation: The user made an invalid change to the "Prefix length" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC191

Changes discarded (short). The invalid change to the input/display characters has been discarded. The input/display delimiter characters must be non-blank and unique. (long).

Explanation: The user made an invalid change to the String delimiter and/or Null column indicator input/display characters on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC192

Changes discarded (short). The invalid change to the Edit locking option has been discarded. Enter one of the listed values for the Edit locking option. (long).

Explanation: The user made an invalid change to the "Edit locking" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC193

Changes discarded (short). The invalid change to the "Optimization for" value has been discarded. Enter a number in the range 0-999999 inclusive. Use 0 if no optimization clause is required. (long).

Explanation: The user made an invalid change to the "Optimization for" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC194

Changes discarded (short). The invalid change to the Keep locks option has been discarded. Enter one of the listed values for the Keep locks option. (long).

Explanation: The user made an invalid change to the "Keep locks" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC195

Changes discarded (short). The invalid change to the Scrollable cursor type option has been discarded. Enter one of the listed values for the Scrollable cursor type option. (long).

Explanation: The user made an invalid change to the "Scrollable cursor type" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor option's panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC196

Changes discarded (short). The invalid change to the Data sampling type option has been discarded. Enter one of the listed values for the Data sampling type option. (long).

Explanation: The user made an invalid change to the "Data sampling type" value on an Editor options panel, and then pressed the PFKey assigned to END (or typed END). The invalid value is discarded and the editor options panel is redisplayed.

User response: No action required. Informational message. After making changes to the options, press ENTER to validate the changes before attempting to exit the panel.

HFMDC197

Enter a positive number (short). A positive number is required for this option when clustered sampling is selected. Enter a number in the range 1-999999999 inclusive. (long).

Explanation: The user selected "clustered sampling" on a Z Data Tools/Db2 Editor options panel, but entered an invalid value for one of the "Include count" field. A positive integer value in the indicated range is required.

User response: Enter a positive integer value in the range 1-999999999 inclusive.

HFMDC198

Invalid value or number (short). A number is required for this option. Valid values are *range* inclusive. (long).

Explanation: The user selected "cluster sampling", or "random sampling" on a Z Data Tools/Db2 Editor options panel, but entered an invalid value in one of the other sampling fields. Valid values for the indicated field are shown in the message *range*.

User response: Enter a value in the range shown in the long message. To access the long message, press the PFKey assigned to HELP (typically PF1), or type HELP on the command line, followed by ENTER.

HFMDC199

Enter a number (short). A number is required for the optimization clause. Valid values are 0-999999 inclusive. Use 0 if no optimization clause is required. (long).

Explanation: The user cleared the "Optimize for" value on a Z Data Tools/Db2 Editor options panel (and pressed ENTER). A value is required for this field.

User response: Enter a value in the range 0-999999 inclusive. Specifying 0 results in no "OPTIMIZE FOR nn ROWS" clause being added to the SELECT statement used to fetch the data for the current editor session. In some situations specifying a small value (eg 1) may change the access path used by Db2® when retrieving the data, thereby improving the response time. See the Db2® SQL Reference manual for a more complete explanation.

HFMDC200

Invalid label range (short). The label range entered is invalid. Press help for more information. (long)

Explanation: The user entered the EXPORT editor primary command, with labels. The labels specified are invalid. The export command cannot be performed.

User response: Check the labels specified carefully. One common cause of this error is specifying a label that does not exist. You can press the PFKey assigned to HELP (PF1 by default) to display additional help for this error.

HFMDC201

NX|X|EX invalid (short). The NX|X|EX parameters cannot be specified when in SNGL mode. Press help for more information. (long).

Explanation: The user entered the EXPORT editor primary command, in SNGL mode, with the NX, X or EX option. The excluded or not-excluded row options are not applicable in SNGL mode. The export command cannot be performed.

User response: Change the display mode to TABL and re-issue the command. Alternatively, remove the excluded or not-excluded option and re-issue the command. You can press the PFKey assigned to HELP (PF1 by default) to display additional help for this error.

HFMDC202

Not in zoom mode (short). The export primary command is invalid in Zoom mode. (long).

Explanation: The user entered the EXPORT editor primary command, in ZOOM mode, which is not supported. The export command cannot be performed.

User response: Return to either TABL or SNGL display modes and re-issue the command. You can press the PFKey assigned to HELP (PF1 by default) to display additional help for this error.

HFMDC203

No labels (short). Labels cannot be specified when in SNGL mode. Press help for more information. (long).

Explanation: The user entered the EXPORT editor primary command, in SNGL mode, with a label range. Labels may only be specified for the EXPORT command in TABL display mode. The export command cannot be performed.

User response: Return to TABL display mode and re-issue the command. You can press the PFKey assigned to HELP (PF1 by default) to display additional help for this error.

HFMDC204

ALL|NX|X|EX missing (short). Labels cannot be specified unless ALL|NX|X|EX is also specified. Press help for more information. (long).

Explanation: The user entered the EXPORT editor primary command, with a label range, but without either the ALL or NX|X|EX parameters. When you specify a label range one of the ALL, NX|X|EX options is required. The export command cannot be performed.

User response: Press the PFKey assigned to HELP (PF1 by default) and review the following topics: "Syntax diagram". ALL|NX|X|EX parameter Modify the command to meet the requirements of the syntax diagram, and the intended usage of the EXPORT command, re-issue the command.

HFMDC206

Export failed (short). The export operation failed. (long).

Explanation: The user issued the EXPORT editor primary command in a Z Data Tools/Db2 editor session. The command failed.

User response: Check the output data set for allocation/space and related errors.

HFMDC207

LRECL mismatch (short). The (RECFM=F) export data set has LRECL=*LRECL 1*. The required LRECL is *LRECL 2*. Specify a different export data set. (long).

Explanation: The user attempted an export operation and specified an LRECL=F data set that is incompatible with the exported data. The export operation cannot be performed.

User response: Use an export data set with LRECL=*LRECL 2*.

HFMDC208

Inconsistent selection (short). The template usage option should only be set to None when the export data format is set to CSV. Select another template usage option. (long).

Explanation: This message is issued by the ZDT/Db2 export function, on the 'To' panel. The user entered '5' (CSV) for the "Template usage" option. However, the export data format option set on the first export options panel is not CSV. The export cannot proceed.

User response: The choice for export data format, on the first export options panel, places restrictions on the template usage option that can be specified on the export 'To' panel. The following list shows which 'template usage' options can be specified for each export data format:

Export data format Template usage option
ZDT/Db2 (SQLDA) format All except 5 (CSV)
Db2 UNLOAD format All except 5 (CSV)
DSNTIAUL format All except 5 (CSV)
User defined All except 5 (CSV)
Delimited variables (CSV) 5 (CSV) only.

HFMDC209

LRECL mismatch (short). The (RECFM=V) export data set has LRECL=*LRECL 1*. The required LRECL is at least *LRECL 2*. Specify a different export data set.

Explanation: The user attempted an export operation and specified an LRECL=F data set that is incompatible with the exported data. The export operation cannot be performed.

User response: Use an export data set with LRECL = *LRECL 2*.

HFMDC220

NX|X|EX invalid (short). The NX|X|EX parameters cannot be specified when in browse mode. Press help for more information. (long).

Explanation: The user entered the EXPORT editor primary command in a browse session, the EXPORT command included a reference to excluded, or not-excluded rows. This is not supported in a Z Data Tools/Db2 browse session.

User response: Modify the EXPORT command to remove any reference to excluded, or not-excluded rows, retry the operation. If the intended usage requires the use of excluded or not-excluded rows as part of the EXPORT command, repeat the process using a Z Data Tools/Db2 view session.

HFMDC221

No labels (short). Labels cannot be specified when in browse mode. Press help for more information. (long).

Explanation: The user entered the EXPORT editor primary command in a browse session, the EXPORT command included label references. There is no prefix area in a Z Data Tools/Db2 browse session, so labels cannot be entered, nor can they be referred to.

User response: Modify the EXPORT command to remove any reference to labels, retry the operation.

HFMDC230

Warning - fixed RECFM (short). The output data set has a fixed record format, which is not recommended when using CSV format. Specify a data set with RECFM=VB or RECFM=V. (long).

Explanation: This message is issued by the ZDT/Db2 Export utility, when CSV format is selected. The attributes of the output data set are inconsistent with CSV format, which produces variable length output records.

User response: In general RECFM=F and RECFM=FB should not be used with CSV format. You can specify the name of a new data set on the export 'To' panel and ZDT/Db2 will calculate the correct LRECL for you.

HFMDC500

Executed OK (short). The *ALTER|COMMENT|CREATE|DROP| GRANT|LABEL|REVOKE* statement executed successfully. Db2® returned an SQLCODE of *SQLCODE*. (long).

Explanation: This message is issued by the ZDT/Db2 Object List and Object Privileges utilities. An SQL statement was executed successfully.

User response: No action required. Informational message.

HFMDC510

No rows returned (short). No rows were fetched, SQL code = +100. (long).

Explanation: This message is issued by the ZDT/Db2 Object List, Object Privileges and Explain utilities functions. A SELECT statement was executed, but returned no rows (SQLCODE+100).

User response: No action required. Informational message.

HFMDC520

No allocation data (short). The PQTY/SQTY values in SYSIBM.SYSTABLEPART indicate that RUNSTATS has not been run. The primary and/or secondary space allocation data for the table space is unavailable. (long).

Explanation: This message is issued by the Alter table space and Alter index panels, accessed from the ZDT/Db2 Object list utility. ZDT/Db2 accessed the Db2® catalog to determine the current space allocation values for the table space or index, however this information is not available because the RUNSTATS utility has not been executed. You can resolve this problem by executing the RUNSTATS utility against the appropriate table space or index, and then repeating the ALTER operation.

User response: No action required. Informational message.

HFMDC521

No allocation data (short). The PQTY/SQTY values in SYSIBM.SYSINDEXPART indicate that RUNSTATS has not been run. The primary and secondary space allocation data for the index space is unavailable. (long).

Explanation: This message is issued by the Alter index panel, accessed from the ZDT/Db2 Object list utility. ZDT/Db2 accessed the Db2® catalog to determine the current space allocation values for the index. However, this information is not available because the RUNSTATS utility has not been executed. You can resolve this problem by executing the RUNSTATS utility against the appropriate index, and then repeating the ALTER operation.

User response: No action required. Informational message.

HFMDC580

Must be blank (short). Specify one value or the other, but not both. (long).

Explanation: This message is issued by the ZDT/Db2 Privileges function. The user selected both options when the two options are mutually exclusive.

User response: Select one option, or the other, but not both. Consult the SQL Reference manual for the relevant syntax diagram for the command.

HFMDC581

Cannot drop directly (short). An auxiliary table cannot be dropped directly. To drop an auxiliary table drop the table that uses the auxiliary table. (long).

Explanation: This message is issued by the ZDT/Db2 "Db2® Object functions" utility. The user specified the combination of 'D' (Drop) and an object type of 12 (Auxiliary table'. This combination is not supported.

User response: You cannot directly drop an auxiliary table. Auxiliary tables should be dropped by a DROP of the table of which the auxiliary table is a part. See the Db2® SQL Reference for a full explanation of the use of auxiliary tables.

HFMDC583

Reserved (short). Sequence number 1 is reserved for the SEL field. Specify a larger value. (long).

Explanation: This message is issued by the "Column Selection/Edit"panel, when displayed from the ZDT/Db2 Object List, Object Privileges and Explain functions (Collectively ISQ functions). The object displays for ISQ functions use a modified version of the ZDT/Db2 editor to display the data, and allow the same template editor functions as a normal ZDT/Db2 editor session; with some restrictions. While you can specify a sequence number in the 'Seq' field, a value of 1 is reserved for the 'Sel' field, which appears on most ISQ panels.

User response: Specify the required ordering starting at '2', instead of '1'.

HFMDC584

Must be blank (short). Specify the ALL parameter, or one or more values, but not both. (long).

Explanation: This message is issued on various Revoke Privilege panels displayed by the ZDT/Db2 Privileges function. When REVOKING a privilege with SYSADM authority you may REVOKE the privilege from ALL users, or list 1-3 individual Db2® authids. These options are mutually exclusive, you cannot select ALL and also enter an individual Db2® authid.

User response: Select ALL if the privilege is to be REVOKEd from all Db2® authids. Alternatively enter one or more individual Db2® authids.

HFMDC585

Must be selected (short). The PUBLIC selection field must be selected when the AT ALL LOCATIONS selection field is selected. (long).

Explanation: This message is issued by the Grant table and Grant column privileges' panels. The user selected the 'AT ALL LOCATIONS' field, but not the 'PUBLIC' field. Selecting the former requires that the latter also be selected

User response: Select both the 'PUBLIC' and 'AT ALL LOCATIONS' fields.

HFMDC586

Not with ROLE (short). The ROLE and PUBLIC options cannot both be selected. PUBLIC is a special Db2® authid, not a valid role name. (long).

Explanation: This message is issued by the Db2® Privileges utility when the user selects both the 'ROLE' and 'PUBLIC' fields. The utility does not support the simultaneous use of both the PUBLIC and ROLE options.

User response: To GRANT/REVOKE the privileges to PUBLIC, clear the ROLE field and process the request. Similarly, to GRANT/REVOKE the privileges to a ROLE, clear the PUBLIC field and process the request.

HFMDC587

Not with ROLE (short). The ROLE and ALL options cannot both be selected. ALL can only be specified with Db2® authids. (long).

Explanation: This message is issued by the Db2® Privileges utility when the user selects both the 'ROLE' and 'ALL' fields. The utility does not support the simultaneous use of both the PUBLIC and ALL options.

User response: To REVOKE the privileges from ALL Db2® authids that possess the privilege, clear the ROLE field and process the request. Similarly, to REVOKE the privileges from a ROLE, clear the ALL field and process the request.

HFMDC588

Not AT ALL LOCATIONS (short). The *Privilege type* privilege cannot be granted to PUBLIC AT ALL LOCATIONS. (long).

Explanation: This message is issued by the Db2® Privileges utility, when granting table or column privileges. The "AT ALL LOCATIONS" may not be used.

User response: Deselect the "AT ALL LOCATIONS" option and process the request.

HFMDC651

Invalid data type (short) ROWID, CLOB, BLOB and DBCLOB columns cannot be specified in a global temporary table definition. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions utility, when attempting to create a global temporary table (GTT). One or more of the columns defined for the GTT has a prohibited data type (ROWID, CLOB, BLOB or DBCLOB). The GTT cannot be created.

User response: Correct the definition for any columns defined as ROWID, CLOB, BLOB or DBCLOB, retry the operation.

HFMDD000

Command list error (short). An error was encountered processing the list of valid commands for panel *panel_name*.

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). The line commands that may be entered against a list of objects are defined using a customization string. An error occurred during the processing of this string for panel *panel_name*.

User response: This is an internal error. Contact HCL Support.

HFMDD001

Customization error (short). An error was encountered processing the template customization string for panel *panel_name*. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). Object lists are displayed using the ZDT/Db2 editor, the template for the object list is modified prior to display. The template customization is defined using a customization string. An error occurred during the processing of this string for panel *panel_name*.

User response: This is an internal error. Contact HCL Support.

HFMDD002

No line commands (short). There are no line commands for this panel. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). In most cases you may enter one or more line commands against each row shown in the object list. In some few cases no line commands are available, in which case this message is issued.

User response: No action required. Informational message.

HFMDD003

VDEFINE error, RC=*Return code* short). A serious ISPF error has been encountered. VDEFINE RC=*Return code*. (long)

Explanation: ZDT/Db2 attempted to use the ISPF VDEFINE service, the call failed with RC = *Return code*. In most cases the current function will terminate.

User response: The causes of this error are:

- Problems with the ISPF environment that have caused the ISPF service call to fail. Contact your Systems Programmer for assistance.
- An internal error. Contact HCL Support.

Points for the Systems Programmer to consider include:

- ISPF libraries allocated incorrectly or incompletely.
- Full ISPF profile data sets.
- Out of memory, or short on memory, conditions. Are other ISPF applications running? Are multiple ISPF logical sessions active?

HFMDD004

ISPF error, RC=*Return code* (short). A serious ISPF error has been encountered in HFM2IPSU. ISPF SELECT SERVICE RC=*Return code* (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor used by the following options: 3.4; 3.5; 4.5). A serious error was encountered in response to an ISPF SELECT service call (return code = *Return code*) In most cases the function will terminate.

User response: The usual causes of this error are:

- Problems with the ISPF environment that have caused the ISPF service call to fail. Contact your Systems Programmer for assistance.
- An internal error. Contact HCL Support.

Points for the System Programmer to consider include:

- ISPF libraries allocated incorrectly or incompletely.
- Full ISPF profile data sets.
- Out of memory, or short on memory, conditions. Are other ISPF applications running? Are multiple ISPF logical sessions active?

HFMDD005

:ISPF error, RC=*Return code* short). A serious ISPF error has been encountered in HFM2IPSU. ISPF CONTROL SERVICE RC=*Return code* long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). A serious error was encountered in response to an ISPF CONTROL service call (return code = *Return code*). In most cases the function will terminate.

User response: The usual causes of this error are:

- Problems with the ISPF environment that have caused the ISPF service call to fail. Contact your Systems Programmer for assistance.
- An internal error. Contact HCL Support.

Points for the Systems Programmer to consider include:

- ISPF libraries allocated incorrectly or incompletely.
- Full ISPF profile data sets.
- Out of memory, or short on memory, conditions. Are other ISPF applications running? Are multiple ISPF logical sessions active?

HFMDD006

VDELETE error, RC=*Return code* (short). A serious ISPF error has been encountered. VDELETE RC=*Return code*(long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). A serious error was encountered in response to an ISPF DELETE service call (return code = *Return code*). In most cases the function will terminate.

User response: The usual causes of this error are:

- Problems with the ISPF environment that have caused the ISPF service call to fail. Contact your Systems Programmer for assistance.
- An internal error. Contact HCL Support.

Points for the Systems Programmer to consider include:

- ISPF libraries allocated incorrectly or incompletely.
- Full ISPF profile data sets.
- Out of memory, or short on memory, conditions. Are other ISPF applications running? Are multiple ISPF logical sessions active?

HFMDD007

No dynamic area (short). A serious error has been encountered. Panel *Panel name* oes not contain a dynamic area and therefore cannot be used to display ISQ editor output. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). A serious error was encountered attempting to display an ISPF panel.

User response: This is an internal error. Contact HCL Support.

HFMDD008

Command not available (short). The SQL command is not supported under CICS®. (long).

Explanation: This message is issued when ZDT/Db2 is running under CICS®, and the user issues the SQL command. The SQL command is not available when running under CICS®.

User response: No action required. Informational message.

HFMDD009

VREP error, RC=RC (short). A serious ISPF error has been encountered. VREPLACE RC=RC. (long).

Explanation: This message is issued by the Basic SQL prototyper. In indicates that a serious ISPF error was encountered.

User response:

The usual causes of this error are:

- Problems with the ISPF environment that have caused the ISPF service call to fail. Contact your Systems Programmer for assistance.
- An internal error. Contact HCL Support.

Points for the Systems Programmer to consider include:

- ISPF libraries allocated incorrectly or incompletely.
- Full ISPF profile data sets.
- Out of memory, or short on memory, conditions. Are other ISPF applications running? Are multiple ISPF logical sessions active?

HFMDD010

Invalid command *Command* (short). "*Command*" is not a valid Z Data Tools Db2® command. *List of valid commands*. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). A command (for example XXXX) was issued on the TSO command line. However, this command is not recognized by the ISQ processor and cannot be run.

User response: No action required. Informational message.

HFMDD011

Too many parms (short). "*Command*"> is followed by one or more parameters. It is only valid without parameters. PARMS=*Parameters*. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). A command *Command*, with parameters, was entered on the TSO command line. *Command* does not require parameters.

User response: Remove the extraneous parameters, re-issue the command.

HFMDD012

No SQL to display (short). There is no SQL statement to display. (long).

Explanation: The user entered the SQL command; however there is no current SQL statement to display.

User response: No action required. Informational message.

HFMDD018

Object not found (short). No Db2® object at location *Location* matches the combination of object details entered at Db2® system *SSID*. Press Help for more information. (long).

Explanation: The user entered a Db2® object name, including location, owner, and name. This object does not exist. You can press PF1 to see additional help for this error.

User response: Press PF1 to see a more detailed explanation of the error. The usual reasons for this error include:

- The nominated object does not exist. Is it spelled correctly?
- When the object name includes lower case letters you must ensure that automatic translation of object names is turned off. See the ZDT/Db2 systems options.
- When the automatic translation of object names is turned off, you must ensure that the object name entered matches the case of the actual object name. For example, DSN81010.EMP matches DSN81010.EMP, but dsn81010.EMP does not.

HFMDD020

Invalid selection (short). The value entered is invalid. Type '?' in the field to display a list of valid values. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). The user entered a line command against a row. However, the line command is not valid for that type of object.

User response: Check for obvious mistakes such as typos. Type '?' in the field and press Enter to show the list of valid commands for the object.

HFMDD022

No SQL to display (short). There is no SQL statement to display. (long).

Explanation: The user entered the SQL command; however there is no current SQL statement to display.

User response: No action required. Informational message.

HFMDD028

Requires Db2® V9 (short). This command can only be issued when connected to a Db2® version 9 (or later) system. (long).

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). When connected to a Db2® version 8 system, the user entered a line command against a row; however the line command is only valid when connected to a Db2® version 9 (or later) system. The command cannot be processed.

User response: No action required. Informational message.

HFMDD029

Must be <= max value (short). The min value entered must be less than or equal to, the max value entered. (long).

Explanation: The user entered a value in a "minimum" field on the ZDT/Db2 Object List Options (2 of 2) panel. The value entered is greater than the corresponding maximum value, which is invalid.

User response: Check the minimum and maximum values for the appropriate column type. The minimum value sets the minimum display width for that column type; the maximum value set the maximum display width for that column type. The minimum display width cannot exceed the maximum.

HFMDD501

SQLCOST = *Cst*>.

Explanation: This message is issued when a SQL statement is about to be executed, and indicates the comparative cost *Cost* of executing the current statement. The cost values are those returned by Db2® when the SQL statement to be executed is prepared. See the SQL Reference manual for the appropriate version of Db2® for a more detailed explanation.

User response: No action required. Informational message.

HFMDD502

Invalid at remote server (short). "*Line ommand*" is only a valid command when entered against an object at the local server. For example, when the location field is blank. (long).

Explanation: The user entered a line command against an object displayed by the ZDT/Db2 Interactive SQL (ISQ) Processor (Options 3.4, 3.5 and 4.5 use the ISQ processor). Some line commands are only valid when ZDT/Db2 is connected to the local Db2® server, an example is privilege related commands such as GRANT and REVOKE. The command cannot be executed.

User response: Connect directly to the remote server and reissue the command.

HFMDD503

Data refreshed

Explanation: This message is issued by the ZDT/Db2 Interactive SQL (ISQ) processor (used by the following options: 3.4; 3.5; 4.5). The data for the currently displayed object list has been refreshed. This can occur when, for example, a line command issued against an entry on the current panel was used to change the entry. One example of this is deleting the object.

User response: No action required. Informational message.

HFMDD504

Invalid command (short). "*Line command*" is not a valid command. Valid commands are D(n), I(n), R(n), and S. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions utility. When a list of, for example, columns is displayed, the user can issue various commands to modify the entries in the list. The command *line command* entered is invalid.

User response: No action required. Informational message.

HFMDD505

Invalid command (short). "*Line command*" is not a valid command. Valid commands are D(n), I(n, and R(n). (long).

Explanation: This message is issued by the ZDT/Db2 Object functions utility. When a list of, for example, columns is displayed, the user can issue various commands to modify the entries in the list. The command *line command* entered is invalid.

User response: No action required. Informational message.

HFMDD506

Invalid selection (short). Use S to select the column. (long).

Explanation: This message is issued by the ZDT/Db2 Object functions utility. An invalid character was entered against an entry in a list of columns.

User response: Use 'S' to select the column.

HFMDD507

Waiting for Db2® prepare …

Explanation: This message is issued by ZDT/Db2 functions that use the ZDT/Db2 editor to display the result table for a SELECT statement. The ZDT/Db2 editor is operating in "Large" mode, meaning that a Db2® scrollable cursor is in use. The two operating modes for the ZDT/Db2 editor are described in detail under Specifying the editor session mode: "normal mode" or "large mode" on page 124. Briefly, when a scrollable cursor is used, there can be a delay while Db2® prepares and opens the SQL statement used to access the data. See also message HFMDD508.

User response: No action required. Informational message.

HFMDD508

Waiting for Db2® open ...

Explanation: This message is issued by those ZDT/Db2 functions that use the ZDT/Db2 editor to display the result table for a SELECT statement. The ZDT/Db2 editor is operating in "Large" mode, meaning that a Db2® scrollable cursor is in use. The two operating modes for the ZDT/Db2 editor are described under Specifying the editor session mode: "normal mode" or "large mode" on page 124. Briefly, when a scrollable cursor is used, there can be a delay while Db2® prepares and opens the SQL statement used to access the data. See also message HFMDD507.

User response: No action required. Informational message.

HFMDD510

Press enter to display the SQL for the view. You can modify/execute the SQL by selecting the "Execute SQL from data set" option.

Explanation: This message is issued when the user enters the VS (show SQL used to create a view) command against a view in an object list display. The Edit/Execute SQL (Data Set) function is used to display the resulting SQL. When the user presses END or CANCEL from the SQL display, the object list is redisplayed.

User response: Press Enter to see the SQL used to create the view.

HFMDD511

Enter the name of a database and table space defined with at least 8K pages. The Db2® Version 8 DSN_FUNCTION_TABLE requires a larger page size than previous Db2® versions.

Explanation: This message is issued when the user attempts to create a function table (DSN_FUNCTION_TABLE) on the Explain Utilities panel. This table requires a table space with a page size of 8K or greater, therefore both a database and table space name are required (The default database/table space will have a page size of 4K, which is inadequate).

User response: Enter the name of a database and table space that have a page size of at least 8K. You may need to contact your Db2® Database/System's Administrator for authority/information on how to do this.

HFMDD541

No RI relationships (short). The related edit (REDIT) command was issued. There are no referential integrity relationships defined on the table. (long).

Explanation: The user entered the REDIT editor primary command. When the object being edited is part of an referential integrity relationship this command displays the other Db2® objects in the relationship. No RI relationships exist that include the object being edited.

User response: No action required. Informational message.

HFMDE495

No Db2® TABLE matches the value entered. It is possible a Db2® object (eg view) that matches the name entered exists - however the selected Db2® utility supports Db2® tables ONLY.

Explanation: This message is issued when a Db2® object name - other than a table - is entered when preparing a Db2® LOAD or UNLOAD utility job. The Db2® LOAD and UNLOAD utilities support only table objects - not views, aliases, synonyms and so on.

User response: Ensure that the Db2® object name specified is a Db2® table.

HFMDJ999

Z Data Tools Db2 component problem - message *Message number*not found in table.

Explanation: This message is issued when an attempt is made to issue message *Message number*, but the message number could not be found. It is a "catch all" message.

User response: If this message occurs during normal processing the likely cause is a logic error. Contact HCL Support for assistance.

Notices

HCL Z Data Tools Licensed Materials - Property of HCL Technologies Ltd. @ Copyright IBM Corp. 2000, 2016. © Copyright HCL Technologies Limited 2017, 2026.

This information was developed for products and services offered in the U.S.A.

HCL may not offer the products, services, or features discussed in this document in other countries. Consult your local HCL representative for information on the products and services currently available in your area. Any reference to an HCL product, program, or service is not intended to state or imply that only that HCL product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any HCL intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-HCL product, program, or service.

For license inquiries regarding double-byte (DBCS) information, contact the HCL Intellectual Property Department in your country or send inquiries, in writing, to:

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel

HCL TECHNOLOGIES LTD. PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. HCL may make improvements and/or changes in the product(s) and/or the program(s) described in this document at any time without notice.

Any references in this information to non-HCL websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this HCL product and use of those websites is at your own risk.

HCL may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

HCL
330 Potrero Ave.
Sunnyvale, CA 94085

USA
Attention: Office of the General Counsel

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by HCL under terms of the HCL Customer Agreement, HCL International Program License Agreement, or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-HCL products was obtained from the suppliers of those products, their published announcements or other publicly available sources. HCL has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-HCL products. Questions on the capabilities of non-HCL products should be addressed to the suppliers of those products.

Statements regarding future direction or intent of HCL Z Data Tools are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to HCL, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. HCL, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. HCL shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp. and/or
HCL Ltd. sample programs.
© Copyright IBM Corp. 2000, 2016. © Copyright HCL Ltd. 2017, 2026.

Programming interface information

This documentation describes intended Programming Interfaces that allow the customer to write programs to obtain the services of Z Data Tools.

Trademarks

HCL, the HCL logo, and hcl.com® are trademarks or registered trademarks of HCL Technologies Ltd., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM® or other companies.

Programming interface information

The User's Guide and Reference documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Z Data Tools.

Index

Special Characters

_ , *See* underscore (_)
? command to list valid prefix commands 165
(1,*), print option 832
(n,m), print option 832
@ , *See* national characters
* , *See* asterisk (*)
\= row selection criteria operator 803
\> row selection criteria operator 803
& row selection criteria connector 802
# , *See* national characters
% , *See* percentage (%)
< row selection criteria operator 803
<= row selection criteria operator 803
= row selection criteria operator 803
> row selection criteria operator 803
>= row selection criteria operator 803
¬< row selection criteria operator 803
¬= row selection criteria operator 803
¬> row selection criteria operator 803
¬BT row selection criteria operator 804
¬IN row selection criteria operator 804
¬LK row selection criteria operator 804
¬NL row selection criteria operator 804
| row selection criteria connector 802
/ (synonym for FIND command) 953
$, *See* national characters

Numerics

132, print option 829
80, print option 829

A

A line command
  databases 610
  functions 710
  indexes 725
  row selection criteria 800
  stored procedures 853
  table edit 869, 882
  table spaces 880
  tables, views and aliases 892
ABOUT primary command 922
action bar
  on ZDT/Db2 panels 38
  pull-down menu, description 445
ADD line command
  tables, views and aliases 893
Additional LISTDEF Statements panel, description 449
Additional TEMPLATE Statements panel, description 450
Advanced SELECT Prototyping panel, description 451
ALL primary command 922
allow
  substitution chars, Db2 LOAD utility option 738
  updates to primary key option 663
alphanumeric columns (Object List Utility) 310
ALS line command
  tables, views and aliases 893
AND row selection criteria connector 802
APF-authorized 27
Application Packages panel, description 469
Application Plan Details panel, description as a details panel 637
Application Plan Privilege Details panel, description as a privilege details panel 768
Application Plan Privileges panel, description as a privileges panel 768
Application Plans panel, description 473
Arbitrary SQL Select Statements 860
arithmetic operators, REXX 263
asterisk (*)
  filtering object lists 309
  using in ZDT/Db2 entry fields 50
AT line command
  columns 504
  tables, views and aliases 893
attention interrupt key 56
attributes
  changing for a column 117
  for a numeric column, changing 107
  for an alphanumeric column, changing 105
Audit trail (option 3.10), description 335
audit trail report
  description 336
  summary statistics 337
Auto detect, compiler language selection option 507
auto-commit count, editor option 673

B

B line command
  DBRMs 608
  row selection criteria 800
  table edit 869, 883
  tables, views and aliases 893
BACKWARD primary command 37, 37, 923
Basic SELECT Prototyping panel, description 476
batch functions
  BATSQL (Batch SQL) 1019
  D2G (Data Generate) 1058
  D2TP (Template create/update) 1060
  DBC (Copy) 1027
  DBI (Import) 1034
  DBP (Print) 1040
  DBX (Export) 1043
batch mode
  JCL syntax 432
  running Z Data Tools/Db2 432
batch mode Db2
  using edit models 433
batch processing
  Audit trail (option 3.10) 250, 335
  Create (option 3.8) 606
  Export (option 3.7) 286
  Import (option 3.6) 256
  Object List (option 3.4) 325, 326
  prerequisite knowledge xiv
  Print (option 3.1) 331
  Utilities (option 3.9) 352, 358
Batch Submission Job Statement Information processing option 818
BATSQL (Batch SQL), ZDT/Db2 batch function 1019
BI line command
  application plans 474
BIGINT data type, description 441
BINARY data type, description 440
Bind Application Plan panel, description 481
Bind Package panel, description 482
BLOB (large object) 444
boolean operators, REXX 265
BOTTOM primary command 37, 924
browse
  materialized query tables 191
Browse
  non-updateable columns 190
  read-only views 190
BROWSE primary command 925
BT line command
  tables, views and aliases 893
BTW 804
build indexes in parallel, Db2 LOAD utility option 739
built-in functions, REXX 266

C

C line command
  row selection criteria 800
  table edit 869, 883
  tables, views and aliases 893
CA-Panvalet
  specifying a data set 52
  using copybooks with non-Db2 data 80
CANCEL primary command 926
canceling long-running Db2 queries 56
CAPS primary command 189, 926
CASE primary command 189, 927
CC line command 801
CDI line command
  tables, views and aliases 893
CH line command
  tables, views and aliases 893
CHANGE
  external function 271
  primary command 928
Change Extended (CX) command 183
changing
  attributes
    alphanumeric column 105
    column 117
    numeric column 107
  long strings 183
  numeric columns 185
  output width 106, 107
  scramble data set name 111
  scramble range values 111
  scramble type 109
  scramble value columns 110
  scramble value option 110
  strings of different length 183
  suppression of leading zeros 107
CHAR data type, description 440
CHARPOS primary command 935
check constraints, create table option 234
Children of Table panel, description 482
choosing your Db2 subsystem 30, 35
CHR line command
  tables, views and aliases 893
CK line command
  tables, views and aliases 893
CLOB (large object) 444
clustered data sampling 170
clustered sampling option 652
COBOL
  compiler
    language selection option 507
    options 820
    return code, maximum allowable 820

specifications 65
compiler, displaying details 29
data description entries 78
replacing options, COBOL processing option 819
COL
line command
indexes 725
tables, views and aliases 893
primary command 934
Collection Privileges panel, description as a privileges panel 768
Collections panel, description 482
column
alphanumeric, changing attributes 105
create attributes, information stored in templates 76
fixed attributes, information stored in templates 76
headings
changing 105
information stored in templates 76
mapping
contiguous columns 118
rules (data type conversion) 118
names, specifying for LOCATE primary command 143
numbers
displaying 167
specifying for LOCATE primary command 143
numeric, changing attributes 107
selecting for attribute change 104
selecting for specifying scrambling options 108
selection criteria, information stored in templates 75
sequence, information stored in templates 76
use attributes, information stored in templates 76
Column Attributes panel (alphanumeric), description 484
Column Attributes panel (DATE), description 489
Column Attributes panel (numeric), description 490
Column Attributes panel (TIME), description 494
Column Attributes panel (TIMESTAMP), description 495
Column Details panel, description as a details panel 636
Column Distribution panel, description 497
column list 147
Column Part Statistics panel, description 497
Column Privilege Details panel, description as a privilege details panel 768
Column Privileges panel, description as a privileges panel 768
column range 147
Column Selection/Edit panel, description 497
Column Selection/Edit primary panel 85, 86, 99
columns
create table option 231
holding for scrolling 100
scrolling to specified name 142
selecting for display 99
Columns in Index panel, description 505
Columns in Table panel, description 505
Columns panel, description 503
COM line command
columns 504
distinct types 638
functions 711
stored procedures 853
tables, views and aliases 893
triggers 904
command line, description 39
commands
continuation 432
reading syntax diagrams xv
commands, Db2, entering from ZDT/Db2 session 200
comment
panels
description 506
list of 506
Comment Column panel, description as a comment panel 506
Comment Distinct Type panel, description as a comment panel 506
Comment Procedure panel, description as a comment panel 506
Comment Specific Function panel, description as a comment panel 506
Comment Table panel, description as a comment panel 506
Comment Trigger panel, description as a comment panel 506
commit when save issued option 673
comparison operators, REXX 261
compiler language selection 65
Compiler Language Selection panel, description 506
compiler, COBOL, displaying details 29
concurrency
and locking options 675
connectors, row selection criteria, list of 802
Constraint Columns for panel, description 509
CONTAINS external function 272
continuation character 432
control statement, continuation 433
convert
row to uppercase 875, 889
spaces to null option 670
uppercase characters in a row to lowercase 872, 886
converting data to uppercase 189
copy
line (row selection criteria) 800
row 869, 883
Copy (option 3.3)
description 246
setting options for the current session 251
Copy Options panel, description 509
Copy Utility (“From”) panel, description 513
Copy Utility (“To”) panel, description 521
COPY Utility (Index Spaces) panel, description 517
COPY Utility (Index Spaces) with LISTDEF panel, description 519
COPY Utility (Table Spaces) panel, description 524
COPY Utility (Table Spaces) with LISTDEF panel, description 526
Copy utility options 68
COPY, generating Db2 utility 354
copybook
coding 295, 301
creating from ZDT/Db2 template 121
Library Management System Exit 52
stored in CA-Panvalet library 52
copying
scrambling data 292
copying data
from a VSAM or QSAM file 251
from an SQL statement 288
from one Db2 object to another 246, 293
from the current ZDT/Db2 editor session 288
scenarios 292
to a VSAM or QSAM file 279
ZDT/Db2 utilities to use 246
CR line command
databases 610
distinct types 638
functions 711
indexes 725
sequences 817
stored procedures 853
synonyms 856
table spaces 880
tables, views and aliases 894
triggers 904
CRA line command
synonyms 856
tables, views and aliases 894
Create (option 3.8), description 244, 497
Create Alias 236
Create Alias panel, description 527
create audit trail option 619, 645
Create Auxiliary Table 243
Create Auxiliary Table panel, description 528
Create Database 223
Create Database panel, description 529
Create Distinct Type 239
Create Distinct Type panel, description 531
Create function
description 239
option list 1/2 240
option list 2/2 240
parameter types 240
parameters 239
returned data type 240
Create Function panel, description 536
Create Function: Option List (1/2) panel, description 533
Create Function: Option List (2/2) panel, description 534
Create Function: Parameter Types panel, description 537
Create Function: Parameters panel, description 539
Create Function: Returned Data Type panel, description 540
Create Index 236
allocation 237
index 238
index column selection 237
index type 237
partitions 238
Create Index panel, description 542
Create Index: Allocation panel, description 545
Create Index: Column Selection panel, description 546
Create Index: Options panel, description 547
Create Index: Partition Values panel, description 550
Create Index: Partitions panel, description 549
Create Index: Type panel, description 552
Create Procedure 240

option list 241
option list 1/2 241
parameter types 241
parameters 241
Create Procedure panel, description 553
Create Procedure: Option List (1/2) panel, description 554
Create Procedure: Option List (2/2) panel, description 556
Create Procedure: Parameter Types panel, description 557
Create Procedure: Parameters panel, description 558
Create Synonym 238
Create Synonym panel, description 560
create table
description 226
table creation options
check constraints 234
columns 231
description 230
generate values 235
nulls and default values 231
options 232
procedure exits 235
referential constraints 232
unique constraints 232
using a model 227
without using a model 230
Create Table panel, description 561
create table space
data
sharing 225
storage 225
locking 225
partitions 226
table space
allocation 225
type 224
Create Table Space panel, description 564
Create Table Space: Allocation panel, description 566
Create Table Space: Data Sharing Options panel, description 567
Create Table Space: Data Storage Options panel, description 568
Create Table Space: Define Partitions panel, description 570
Create Table Space: Locking Options panel, description 572
Create Table Space: Type panel, description 573
Create Table: Column Check Constraints panel, description 575
Create Table: Column Referential Constraints panel, description 576
Create Table: Columns panel, description 578
Create Table: Generate Values panel, description 580
Create Table: Model Load panel, description 582
Create Table: Nulls and Default Values panel, description 584
Create Table: Options panel, description 587
Create Table: Procedure Exits panel, description 589
Create Table: Table Check Constraints panel, description 590
Create Table: Table Referential Constraint panel, description 592
Create Table: Table Referential Constraints panel, description 593
Create Table: Unique Constraints panel, description 595
Create trigger
description 241
details 242
search conditions 242
SQL statement 242
Create Trigger panel, description 596
Create Trigger: Details panel, description 598
Create Trigger: Search Condition panel, description 600
Create Trigger: SQL statement panel, description 600
Create View 235
Create View panel, description 601
creating
copybook from ZDT/Db2 template 121
data for a column 118
CRS line command
databases 610
CRT line command
table spaces 880
CRX line command
tables, views and aliases 894
CS line command
tables, views and aliases 894
cursor stability option 678
CX (Change Extended) command 183

D

D
edit prefix command 869, 883
line command
EXPLAIN plan table 754
indexes 725
row selection criteria 801
table spaces 880
tables, views and aliases 894
table space partition prefix command 571
D line command
storage groups 849
D2G (Data Generate), ZDT/Db2 batch function 1058
D2TP (Template create/update), ZDT/Db2 batch function 1060
data
copying 246
displaying data in hexadecimal format 165
format, output from Export (option 3.7) 285
large objects (LOBs) 444
sampling 170
sorting in editor session 164
types, VARCHAR 178
Data Create Utility panel, description 603
Data description support
HLASM
DS/DC 82
DSECT 82
PL/I
not supported 82
REFER 82
VARYING 82
VARYINGZ 82
data format, export options 689
data set
name, specifying on panel 52
Data set DISP, print option 830
Data set name, print option 831
Data set, print option 828
data sharing environment 625
data type
displaying 167
rules, conversion 118
data types, description
BIGINT 441
BINARY 440
CHAR 440
DATE 442
DECFLOAT 441
DECIMAL 441
DOUBLE 442
FLOAT 442
GRAPHIC 440
INTEGER 441
NUMERIC 441
REAL 441
ROWID 442
SMALLINT 441
TIME 442
VARBINARY 440
VARCHAR 440
VARGRAPHIC 440
data, displayed
changing view 152
limiting
excluding rows 162
row labels 150
database
for a Db2 object, specifying 49
Database Details panel, description as a details panel 636
Database Privilege Details panel, description as a privilege details panel 768
Database Privileges panel, description as a privileges panel 768
Database Request Modules
panel, description 607
Database Structure panel, description 609
Databases panel, description 609
DATE data type, description 442
Db2
commands, entering from ZDT/Db2 session 200
data
sharing environment 625
types supported 440
Interactive, interface to 438
object name, specifying on panel 46
objects
creating and dropping 208
working with lists of 303
privileges, managing 339
releases supported by ZDT/Db2 24
subsystem
changing using Db2 SSID field 30
changing using SSID command 32
selecting 30
Subsystem Selection menu 33
table, printing contents 328
utilities
COPY 354
generating 346
LOAD 357
REBUILD 365
RECOVER 368
REORG 371
RUNSTATS 376
UNLOAD 294, 380
Db2 Browse panel, description 612
Db2 Edit panel, description 616

Db2 Location Selection panel, description 621
Db2 Object Functions panel, description 622
Db2 Save Error Action panel, description 624
Db2 SSID field, changing Db2 subsystem 30
Db2 Subsystem Selection panel, description 624
Db2 tables
  populating 244
Db2 Template Description panel, description 626
Db2 Utilities panel, description 627
Db2 View panel, description 631
Db2I Primary Option Menu panel, description 635
DB2SYS command , *See* SSID command, changing Db2 subsystem
DBC (Copy), ZDT/Db2 batch function 1027
DBCLOB (large object) 444
DBCS
  data, Db2 LOAD utility option 740
  displaying or editing 70
  Settings: Load utility options (option 0.5) 740, 740
DBCS processing option 821
DBI (Import), ZDT/Db2 batch function 1034
DBP (Print), ZDT/Db2 batch function 1040
DBRM Details panel, description as a details panel 637
DBX (Export), ZDT/Db2 batch function 1043
DD line command
  row selection criteria 801
  table edit 869, 883
DECFLOAT data type, description 441
DECIMAL data type, description 441
decimal data, export options 696
DELCOL primary command 935
delete
  existing rows, option 512
  line (row selection criteria) 801
  row 869, 883
  row (partition information) 571
DELETE primary command 936
DEP line command
  packages 471
  sequences 817
DESCRIBE primary command 938
description, adding to a template 85
DESELECT primary command 938
details panels
  description 636
  list of 636
DI line command
  columns 504
display
  excluded row 870, 884
  format
    examples 153
    single (SNGL) 156
    table (TABL) 153
    using Zoom function key 160
Display Format for DATE/TIME Columns processing option 861
display prefix area option 657
Display Row panel, description 637
display, print output, when browsing or editing 153
displaying
  privileges 342
Distinct Type Details panel, description as a details panel 637
Distinct Type Privileges panel, description as a privileges panel 768
Distinct Type Usage Privilege Details panel, description as a privilege details panel 768
Distinct Types panel, description 637
DK line command
  for EXPLAIN plan table 755
  for EXPLAIN statement table 845
do not set COPY pending flag, Db2 LOAD utility option 738
DOUBLE data type, description 441
DOUBLE PRECISION data type, description 441
DOWN primary command 37, 939
DP line command
  for EXPLAIN plan table 755
  for EXPLAIN statement table 846
DQ line command
  for EXPLAIN plan table 755
  for EXPLAIN statement table 846
DR line command
  databases 610
  distinct types 638
  functions 711
  indexes 725
  sequences 817
  stored procedures 853
  synonyms 856
  table spaces 880
  tables, views and aliases 894
  triggers 904
drop
  function 243
  panels
    description 640
    list of 640
Drop Alias panel, description as a drop panel 640
Drop Confirmation panel, description 640
Drop Database panel, description as a drop panel 640
Drop Distinct Type panel, description as a drop panel 640
Drop Function panel, description as a drop panel 640
Drop Index panel, description as a drop panel 640
Drop Stored Procedure panel, description as a drop panel 640
Drop Synonym panel, description as a drop panel 640
Drop Table panel, description as a drop panel 640
Drop Table Space panel, description as a drop panel 640
Drop Trigger panel, description as a drop panel 640
Drop View panel, description as a drop panel 640
DROP, RETURN return value 278
DS line command
  databases 611
  table spaces 881
DS/DC 82
DSECT 82
DT line command
  columns 504
  schemas 813
duplicate key processing, option 511

E

E line command
  table edit 870, 883
  tables, views and aliases 894
Edit (option 2)
  converting data to uppercase 189
  displaying data in hexadecimal format 165
  errors, handling 202
  limiting data displayed
    excluding rows 162
  scrolling 160
  sorting data 164
  VARCHAR, options for processing 178
  varying-length columns, options for processing 178
  zooming 158
edit models
  example 434
edit models Db2
  inserting in JCL 433
Edit options 67
edit prefix commands, list of 869, 882
EDIT primary command 940
edit related table 874, 887
Edit/Execute SQL (Data Set) panel, description 641
EDITCOL primary command 941
editing
  templates 84
editor
  finding
    string 146
  locating
    row 142
  locating a labeled row 142
editor CAPS Setting 645
Editor Options (1 of 8) panel, description 644
Editor Options (2 of 8) panel, description 649
Editor Options (3 of 8) panel, description 655
Editor Options (4 of 8) panel, description 662
Editor Options (5 of 8) panel, description 667
Editor Options (6 of 8) panel, description 672
Editor Options (7 of 8) panel, description 675
Editor Options (8 of 8) panel, description 680
editor session
  data, displayed, changing view 152
  finding
    numeric column 152
  large numeric fields 168
  large table options 170
  limiting data displayed
    row labels 150
  locating
    column 142
  specifying type 124
  SQL optimization options 169
  start position, specifying 126
  Unicode data 179
Encapsulation of SQL Identifiers in Double Quotes processing option 858
end of string delimiter
  display option 669
  input option 669
END primary command 942
ending a session
  ZDT/Db2 editor 133
enforce constraints, Db2 LOAD utility option 738
engineering, reverse 324
Enter and Execute Db2 Commands panel, description 683
Enter key and commit processing options 648

Enter, Execute and Explain SQL Statements panel, description 684
EQ row selection criteria operator 803
errors
in an Edit session, handling 202
SQL statements 427
EXCLUDE primary command
description 943
exclude row from display 876, 889
excluded rows
sorting in editor session 164
using RESET to redisplay 994
excluding rows 162
EXECUTE primary command 948
execution, export options 690
Exit 29
EXIT statement 268
EXPAND primary command 949
EXPLAIN
interpretation panel, description 686
plan table primary commands
D 755
DK 755
DP 755
DQ 755
FR 756
I 756
IH 756
K 756
M 757
P 757
SR 757
T 757
UH 758
X 758
primary command 950
statement table primary commands
DK 845
DP 846
DQ 846
I 846
K 847
M 847
P 847
Explain Utilities panel, description 686
export
options
batch data set creation 691
data format 689
decimal data 696
execution 690
floating point data 697
graphic data 697
indicator 694
integer data 697
placement 693
type 693
usage 693
Options (1 of 3) panel, description 689
Options (2 of 3) panel, description 692
Options (3 of 3) panel, description 696
Export “To” panel, description 701
Export (option 3.7)
data formats 285
description 279
notes on using 287
running in batch 294
scenarios 292
setting options for the current session 282
export data set, export options 691
EXPORT primary command
description 950
Export utility options 69
Export Utility panel, description 698
exporting
scrambling data 292
expressions, REXX 258
external functions provided by ZDT/Db2
CHANGE 271
CONTAINS 272
FLD 273
NCONTAIN 274
PRINT 274
PRTCOUNT 275
RECSIN 275
RECSOUT 275
TALLY 276
TM 277
WRITE 277

F

F line command
schemas 813
table edit 870, 884
Field Selection/Edit panel, description 704
field-level help 60
FILE primary command 953
filter line, object lists 309
Find Extended (FX) command 146
FIND primary command
description 953
Object List utility 314
finding
long strings 146
numeric column 152
string 146, 314
first page, scrolling to 37
FIRST parameter, using in LOCATE command 145
FK line command
tables, views and aliases 894
flavors, of Z Data Tools xiii
FLD
external function 273
parameter, using in LOCATE command 144
FLOAT data type, description 441
floating point data, export options 697
FMAP primary command 961
foreign key
editor options 664
multiple 665
Objects (option 3.2) 233, 234, 577, 583, 594
scale line, displaying 656
system options 862
Foreign Keys for Table panel, description 708
FORMAT primary command 962
format, examples 153
FORWARD primary command 37, 37, 963
FR line command (for EXPLAIN plan table) 755
fragments, syntax diagrams xv
FRE line command
application plans 474
packages 471
Free Application Plan panel, description 708
Free Package panel, description 708
FREE primary command 964
FS
line command 870, 884
primary command 965
FT primary command 966
Function Details panel, description as a details panel 637
function keys
default settings 39
Zoom 160
Function Privilege Details panel, description as a privilege details panel 768
Function Privileges panel, description as a privileges panel 768
functions
batch 1019
built-in, REXX 266
external, provided by ZDT/Db2
CHANGE 271
CONTAINS 272
FLD 273
NCONTAIN 274
PRINT 274
PRTCOUNT 275
RECSIN 275
RECSOUT 275
TALLY 276
TM 277
WRITE 277
Functions panel, description 709
functions, ZDT/Db2
BATSQL (Batch SQL) 1019
D2G (Data Generate) 1058
D2TP (Template create/update) 1060
DBC (Copy) 1027
DBI (Import) 1034
DBP (Print) 1040
DBX (Export) 1043
description 1018
FX (Find Extended) command 146

G

G line command
application plans 474
collections 483
columns 504
databases 611
distinct types 638
functions 711
packages 471
sequences 817
storage group usage privileges 851
stored procedures 853
table edit 870, 884
table spaces 881
tables, views and aliases 894
GE row selection criteria operator 803
GEN line command
databases 611
distinct types 639
functions 711
indexes 725
reverse engineering 324
schemas 813
stored procedures 854
table spaces 881
tables, views and aliases 894
Generate SQL From Db2 Catalog panel, description 712
get row from clipboard 870, 884
Grant Application Plan Privileges panel, description as a grant privileges panel 715
Grant Buffer Pool Privileges panel, description as a grant privileges panel 715
Grant Collection Privileges panel, description as a grant privileges panel 715

Grant Column Privileges panel, description as a grant privileges panel 715
Grant Database Privileges panel, description as a grant privileges panel 715
Grant Distinct Type Privileges panel, description as a grant privileges panel 715
Grant Function Privileges panel, description as a grant privileges panel 715
Grant Package Privileges panel, description as a grant privileges panel 715
grant privileges panels
  description 715
  list of 715
Grant Schema Privileges panel, description as a grant privileges panel 715
Grant Sequence Privileges panel, description as a grant privileges panel 715
Grant Storage Group Privileges panel, description as a grant privileges panel 715
Grant Stored Procedure Privileges panel, description as a grant privileges panel 715
Grant System Privileges panel, description as a grant privileges panel 716
Grant Table Privileges panel, description as a grant privileges panel 716
Grant Table Space Use Privileges panel, description as a grant privileges panel 716
granting privileges 342
graphic data, export options 697
GRAPHIC type, description 440
GT 803

H

H line command
  columns 504
help
  accessing 57
  fields 60
  menu 448
  tutorial 57
HEX primary command 966
hexadecimal, displaying data 165
HFMDB2 432
HIDE primary command 967
HINT primary command 968
HLASM
  compiler
    specifications 66
  data description entries 78
  data description support
    DS/DC 82
    DSECT 82
HLASM Processing Options 821
HLASM, compiler language selection option 508
HOLD primary command 968
holding columns for scrolling 100

I

I line command
  application plans 474
  columns 504
  databases 611
  DBRMs 608
  distinct types 639
  for EXPLAIN plan table 756
  for EXPLAIN statement table 846
  functions 711
  indexes 725
  packages 471
  plan table rows panel 759
  storage group usage privileges 851
  stored procedures 854
  synonyms 856
  table edit 870, 884
  table space partition prefix command 570
  table spaces 881
  tables, views and aliases 894
  triggers 904
ICS line command
  databases 611
  indexes 725
  table spaces 881
ignore RI/constraint errors, option 512
IH line command (for EXPLAIN plan table) 756
Import (option 3.6) 251, 279
  description 251
  REXX procedure, using 257
  scenarios 292
Import Options panel, description 716
importing
  scrambling data 292
IN row selection criteria operator 804
include count option 653
index
  information, editor options 665
Index Details panel, description as a details panel 636
Index Parts panel, description 723
Indexes and Columns for panel, description 726
Indexes panel, description 723
INDEXINF primary command 970
indicator, export options 694
information
  stored in a template 75
initial display option 645
initial skip count option 653
insensitive cursor option 651
insert
  primary key 872, 885
  row 870, 884
  row (partition information) 570
INSRPT primary command 971
INTEGER data type
  description 441
integer data, export options 697
interpreting errors in an Edit session 202
IP edit prefix command 871, 885
ISPF
  familiarization with xiv
  settings 67
ISPF Settings panel, description 727

J

JCL
  running Z Data Tools/Db2
    batch mode 432
Job card specifications 65

K

K line command
  EXPLAIN plan table 756
  EXPLAIN statement table 846
  plan table rows panel 759
Key, index and export options 647
keys, attention interrupt 56
keywords, syntax diagrams xv

L

L line command
  table edit 872, 886
LAB line command
  columns 504
  tables, views and aliases 895
label
  editor-assigned 151
  row 876, 889
labeled rows
  locating 142
language selection, compiler language selection option 507
large
  numeric fields 168
  objects (LOBs) 444
  table options 170
large mode editor session, specifying 124
last page, scrolling to 37
LAST parameter, using in LOCATE command 145
launching ZDT/Db2 26
LC line command
  table edit 872, 886
LCC line command, table edit 873, 886
LE row selection criteria operator 803
leading zeros, showing 107
left justify numerics, editor option 660
LEFT primary command 971
LHEX primary command 973
limiting data 147
limiting data in editor sessions
  excluding rows 162
  row labels 150
line (row selection criteria)
  copy 800
  delete 801
  move 801
  repeat 802
line command
  area (Object List Utility) 314
  query (Object List Utility) 320
Lines per page, print option 831
LISTCAT
  Output Display panel, description 729
LISTDEF Options panel, description 729
LISTDEF statement, generating in Db2 utility job 350
listdef utility
  options 69
LK row selection criteria operator 804
LOAD
  from panel, description 731
  generating Db2 utility 357
  options 69
  setting options for the current session 363
LOAD Utility - Using Templates panel, description 735
LOAD Utility Options panel, description 737
LOAD Utility panel using Db2 templates panel, description 734
LOAD Utility panel, description 733
LOB
  description 444
LOBBRWS primary command 974
LOBEDIT primary command 975
LOBLEN primary command 976
LOBVIEW primary command 977
LOCATE primary command
  description 978
  in editor sessions 142, 142
  Object List utility 314
locating
  column 142, 314
  label (assigned to a row) 142
  labeled row 142

row 142, 314
location of a Db2 object, specifying 47
locking
copy option 510
log changes, Db2 LOAD utility option 738
logical operators, REXX 265
long strings
changing 183
finding 146
long-running Db2 queries, canceling 56
LPT line command
application plans 474
packages 471
LT 803
LZ primary command , *See* LZERO primary command
LZERO primary command 982

M

M
edit prefix command 873, 886
line command
application plans 474
EXPLAIN plan table 756
EXPLAIN statement table 847
row selection criteria 801
Manage Db2 Privileges panel
description 741
using 340
mapping
automatically by ZDT/Db2 113
contiguous columns 118
data using templates 112
specifying 114
templates, where you can use 113
Materialized Query Table Details panel, description as a details panel 636
materialized query tables
updating 191
maximum discards, Db2 LOAD utility option 740
maximum return code, compiling a COBOL copybook 820
maximum return code, compiling a PL/I copybook 827
member name
PDS, specifying on panel 52
specifying on panel 52
mixed DBCS data, Db2 LOAD utility option 740
MM line command
row selection criteria 801
table edit 873, 887
MOD, print option 830
mode of editor session, specifying 124
model table, using for Create Table utility 227
move
line (row selection criteria) 801
row 873, 886

N

name of a Db2 object, specifying 49
national characters xiv
Native unicode processing, option 512, 690
NBT row selection criteria operator 804
NCONTAIN external function 274
NE row selection criteria operator 803
NEW primary command 983
NEWS primary command 983
NEXT
parameter, using in LOCATE command 144
primary command 983
NEXTRPT primary command 984
NIN row selection criteria operator 804
NL row selection criteria operator 804
NLK row selection criteria operator 804
NNL row selection criteria operator 804
no with clause option 678
NOALIGN processing option 822
non-Db2 data, using templates with 77
non-excluded rows, sorting in editor session 164
non-updateable columns, edit 190, 191
normal mode editor session, specifying 124
NR row selection criteria operator 804
null column input indicator
exporting in data 301
option 670
nullable columns, exporting data containing 301
nulls and default values, create table option 231
number of index keys, Db2 LOAD utility option 739
numeric
column, finding 152
columns (Object List Utility) 310
fields, large 168
NUMERIC data type, description 441
numerics, left justify
table display format 660

O

O line command
schemas 813
O prefix command 873
object
name, specifying on panel 46
Object List (option 3.4) 303
description 303
filter line 309
finding string 314
GEN line command 324
limiting data displayed 309
line command
area 314
query 320
locating a row or column 314
operator line 309
primary commands
FIND 314
LOCATE 314
SORT 323
reverse engineering 324
ROW line command 322
sort
multiple columns 324
one column 323
valid commands 320
working with object list panels 307
Object List Line Commands panel, description 742
Object List Options 68
Object List Utility panel, description 748
Objects (option 3.2), description 208
OLD, print option 830
On prefix command 873
only commit when no save errors option 673
OO prefix command 873
operator line, object lists 309
operators, row selection criteria, list of 803
optimization options, SQL 169
Optimization, large table and data sampling options 647, 648
option panels 62
option utility
options 70
options
create table 230
create table option 232
export, setting 282
menu 446
selecting on ZDT/Db2 panels 45
ZDT/Db2 editor, setting 126
OPTIONS Options panel, description 750
OR row selection criteria connector 802
ORDER primary command 984
Output class, print option 831
output data set allocation, option 67
output destination, print option 828
Output destination, print option 828
output width, changing 106, 107
overlaying rows 873
overview
templates 21, 74
ZDT/Db2 functions 21
owner of a Db2 object, specifying 48

P

P line command
application plans 474
collections 483
columns 505
databases 611
distinct types 639
EXPLAIN plan table 757
EXPLAIN statement table 847
functions 711
packages 471
schemas 813
storage groups 850
stored procedures 854
table
edit 874, 887
spaces 881
tables, views and aliases 895
PA line command
tables, views and aliases 895
Package Dependencies panel, description 753
Package Details panel, description as a details panel 637
Package List panel, description 753
Package Privilege Details panel, description as a privilege details panel 768
Package Privileges panel, description as a privileges panel 768
Page skip, print option 830
pages, scrolling to
first 37
last 37
panel
body 39
record count 39
title 39
panels
Column Selection/Edit 85, 85, 86, 99, 99
Export Options (1 of 3) panel, description 689
Export Options (2 of 3) panel, description 692
Export Options (3 of 3) panel, description 696
ISPF Settings panel 836
privileges, managing 339
selecting options 45

working with 38
ZDT/Db2 Object List Options (1 of 2) panel, description 743
ZDT/Db2 Object List Options (2 of 2) panel, description 746
ZDT/Db2 System Options (1 of 4) panel, description 857
ZDT/Db2 System Options (2 of 4) panel, description 860
ZDT/Db2 System Options (3 of 4) panel, description 863
ZDT/Db2 System Options (4 of 4) panel, description 864
PAR line command
tables, views and aliases 895
Parents of Table panel, description 753
PARM line command
functions 711
stored procedures 854
Partition Values Option 238
Partitioned Table Statistics panel, description 753
patterns, using in ZDT/Db2 entry fields 50
PB primary command
description 986
using 333
percentage (%)
filtering object lists 310
using in ZDT/Db2 entry fields 50
performance
general tips 1018
PK line command
tables, views and aliases 895
PKG line command
application plans 474
collections 483
functions 711
indexes 725
stored procedures 854
tables, views and aliases 895
triggers 904
PKL line command
application plans 474
collections 483
packages 471
PL line command
collections 483
DBRMs 608
indexes 725
packages 471
tables, views and aliases 895
PL/I
compiler
language selection option 507
options 826
return code, maximum allowable 827
specifications 66
data description support
REFER 82
VARYING 82
VARYINGZ 82
Data description support
not supported 82
DECLARE statements 79
placement, export options 693
Plan Dependencies panel, description 753
Plan Enabled/Disabled Connections panel, description 754
PLAN primary command 986
Plan Table Rows panel, description 754
populating a Db2 table with data 244
positioning prefix area 165
PR line command
tables, views and aliases 895
predicate
specifying 394
prefix
area
listing valid commands 165
using commands in 186
commands
list of 869, 882
listing valid 165
row selection criteria, list of 800
using 186
prefix area
setting position 165
prefix area option 657
prefix commands
O 873
On 873
OO 874
prefix length option 657
PREFIX primary command 165, 187, 987
preformat unused pages, Db2 LOAD utility option 737
prerequisite information, for ZDT/Db2 xv
Preserve copybook library option 508
PREV parameter, using in LOCATE command 145
PREVIOUS primary command 988
PREVRPT primary command 988
primary commands
ABOUT 922
ALL (SQL prototyping) 922
BACKWARD 37, 37, 923
BOTTOM 37, 924
BROWSE 925
CANCEL 926
CAPS 926
CAPS (editing) 189
CASE 927
CASE (editing) 189
CHANGE 928
CHARPOS 935
COL 934
DELCOL 935
DELETE 936
DESCRIBE 938
DESELECT 938
DOWN 37, 939
EDIT 940
EDITCOL 941
END 942
EXCLUDE 943
EXECUTE 948
EXPAND 949
EXPLAIN 950
EXPLAIN plan table
D 755
DK 755
DP 755
DQ 755
FR 756
I 756
IH 756
K 756
M 757
P 757
SR 757
T 757
UH 758
X 758
EXPLAIN statement table
DK 845
DP 846
DQ 846
I 846
K 846
M 847
P 847
EXPORT 950
FILE 953
FIND 953
FMAP 961
FORMAT 962
FORWARD 37, 37, 963
FREE 964
FS 965
FT 966
HEX 966
HIDE 967
HINT 968
HOLD 968
INDEXINF 970
INSRPT 971
LEFT 971
LHEX 973
LOBBRWS 974
LOBEDIT 975
LOBLEN 976
LOBVIEW 977
LOCATE 142, 978
NEW 983
NEWS 983
NEXT 983
NEXTRPT 984
object list
sort multiple column 324
sort one column 323
Object List
SORT 323
ORDER 984
PB 986
PLAN 986
PREFIX 987
PREVIOUS 988
PREVRPT 988
PRINT 989
PROTCOL 990
PURGE 990
QUIT , *See* CANCEL primary command
RCHANGE 991
RD 991
REDIT 992
REFRESH 993
REFS 994
RESET 994
RESET (in Edit sessions) 187
RFIND 995
RIGHT 996
RP 997
RUNTEMP 998
SAVE 999
SAVEAS 999
SHADOW 1000
SHOW 34, 1001, 1001
SHOWCOB 1003
SHOWPLI 1003
SORT 1003
SORT (Object List) 323
SQL 1005
SQLID 1006

SSID 32
TEDIT 152, 1006
TOP 37, 1007
TP 1007
TYPE 1008
UNDO 1008
UP 37
VER 1010
VIEW 1011
WIDTH 1011
XMLBRWS 1013
XMLEDIT 1014
XMLLEN 1015
XMLVIEW 1015
ZOOM 158, 1016
primary key
insert 872, 885
Primary Key for Table panel, description 760
Primary Option Menu
Create (option 3.8) 244
panel description 760
Set Processing Options (option 0) 62
Settings (option 0) 833, 836
SQL prototyping, execution and analysis (option 4) 390
Start Db2 Interactive (option 5) 438, 438
Utilities (option 3.9) 346
PRINT
external function 274
primary command 989
Print (option 3.1) 328, 331
Print Audit Trail panel, description 761
Print Browse (option 3.11)
description 333
primary commands
FIND 953
LOCATE 978
PRINT 989
PURGE 990
Print settings 64
Print utility options 68
Print Utility Options panel, description 763
Print Utility panel, description 764
printing
audit trail report 335
contents of a Db2 table 328
looking at the output 333
rows from a Db2 table 331
template 77
privilege, for object, revoking
list of panels 768
privileges
description 768
display panels, format 769
displaying 342
functions 339
grant panels, format 716
granting 342
panels, list of 768
revoke panels, format 799
revoking 344
using the Manage Db2 Privileges panel 340
Privileges (option 3.5) 339
description 339
Procedure Details panel, description as a details panel 637
procedure exits, create table option 235
Process menu 445
processing options
batch
Batch Submission Job Statement Information 818
COBOL
additional SYSLIB data sets 819
compiler specifications 65
replacing options 819
compiler
auto detect 508
COBOL 507, 820
HLASM 508
language selection 65, 507
PL/I 507, 826
copy
Copy Options panel 251
Delete existing rows 512
duplicate key processing 511
ignore RI/constraint errors 512
locking 511
Native unicode processing 512
use uncommitted read 510
data set allocation
output 67
temporary 66
Edit options 67
edit, create audit trail 619, 645
editor
allow updates to primary key 663
Auto-commit count 674
clustered sampling 652
commit when save issued 673
concurrency and locking 675
Concurrency and locking options 648
convert spaces to nulls 670
cursor stability 678
display prefix area 657
editor CAPS Setting 645
end of string delimiter (display) 670
end of string delimiter (input) 669
Enter key and commit processing options 648
include count 653
initial display 645
initial skip count 653
insensitive cursor type 651
Key, index and export options 647
no with clause 678
null column input indicator 670
Only commit when no save errors 673
Optimization, large table and data sampling options 647
prefix area 657
prefix length 657
random sampling 652, 653
read stability 678
read stability keep locks 678
remove trailing spaces 669
repeatable read 678
repeatable read keep locks 679
row count 614, 617, 633
sampling frequency 654
sampling seed 654
sampling type 652
scrollable cursor type 651
sensitive cursor type 651
Show end of string 668
show excluded shadow lines 661
Show export options 666
show foreign key 664
show index indicators 665
show prefix area on the right 658
show primary key 663
show SQLCODE 658
single display format options 659
skip count 653
table display format 656
Table/Single format, prefix area and shadow options 647
Use row count value 650
Use/Keep exclusive locks 679
Use/Keep share locks 679
Use/Keep update locks 679
Varying length and nullable column options 648
export
Native unicode processing 690
Export utility options 69
global, setting for ZDT/Db2 833
HLASM
DBCS 822
NOALIGN 822
ISPF settings 67, 836
Job card specifications 65
listdef utility options 69
load
allow substitution chars 738
build indexes in parallel 739
DBCS data 740
do not set COPY pending flag 738
Enforce constraints 738
log changes 738
maximum discards 740
mixed DBCS data 740
number of index keys 739
preformat unused pages 738
replace existing data 739
resume (append new data) 739
reuse existing data 739
SBCS data 739
use IEEE floating point 738
Object List Options 68
option utility options 70
PL/I
additional SYSLIB data sets 826
compiler specifications 66, 66
print
(1,*) 832
(n,m) 832
132 829
80 829
Data set 829
Data set DISP 830
Data set name 831
Lines per page 832
MOD 830
OLD 830
Output class 831
output destination 828
Output destination 828
Page skip 830
Record length 829
Record limits 832
REXX 829
SYSPRINT 828
Terminal 828
Translate non-printable chars 831
Uppercase message text 831
Wide print 830
Print Processing Options 68, 68, 69
Print settings 64
setting default 62
system options 64
template utility options 70

trace options 67
unload utility options 70
utilities 68
View options 67
ZDT/Db2 system
Arbitrary SQL Select Statements 860
Display Format for DATE/TIME Columns 861
Encapsulation of SQL Identifiers in Double Quotes 858
Retrieve foreign key information when building templates 862
Translate Db2 object names 857
Translate input SQL statements 858
Use uncommitted read when accessing the Db2 catalog 861
ZDT/Db2 Object List Options (1 of 2) panel 743
ZDT/Db2 Object List Options (2 of 2) panel 746
ZDT/Db2 System Options (1 of 4) panel 857
ZDT/Db2 System Options (2 of 4) panel 860
ZDT/Db2 System Options (3 of 4) panel 863
ZDT/Db2 System Options (4 of 4) panel 864
ZDT/Db2 system options 65
PROTCOL primary command 990
prototyping, primary commands
ALL 922
CANCEL 926
DESELECT 938
EXECUTE 949
INSRPT 971
NEXTRPT 984
PREVRPT 988
RESET 994
SQL 1005
UNDO 1008
PRTCOUNT external function 275
PS line command
distinct types 639
functions 711
stored procedures 854
triggers 904
PT line command
indexes 725
storage groups 850
table spaces 881
PTF level, displaying ZDT/Db2 27
pull-down menus action bar 38
PURGE primary command 990

Q

QUIT primary command , *See* CANCEL primary command

R

R
edit prefix command 874, 887
line command
application plans 475
collections 484
databases 611
distinct types 639
functions 711
packages 471
row selection criteria 802
schemas 813
stored procedures 854
table spaces 881
tables, views and aliases 895
xxx 475
table space partition prefix command 570
R line command
storage group usage privileges 852
random data sampling 172
random sampling option 652, 653
RBI line command
application plans 475
packages 471
RCHANGE primary command 991
RD primary command 991
RDT line command
functions 712
RE edit prefix command 874, 887
read
stability keeplocks option 678
stability option 678
read-only views 190
REAL data type, description 441
REBUILD (Indexes for Table Spaces) panel, description 773
REBUILD (Indexes for Table Spaces) with LISTDEF panel, description 775
REBUILD (Indexes) with LISTDEF panel, description 779
REBUILD Utility (Indexes) panel, description 777
REBUILD, generating Db2 utility 365
Record length, print option 829
Record limits, print option 832
RECOVER Utility (Index Spaces) panel, description 780
RECOVER Utility (Index Spaces) with LISTDEF panel, description 782
RECOVER Utility (Table Spaces) panel, description 783
RECOVER Utility (Table Spaces) with LISTDEF panel, description 785
RECOVER, generating Db2 utility 368
Recovery Information panel, description 787
RECSIN external function 275
RECSOUT external function 275
REDIT primary command 992
REFER 82
referential constraints
create table option 232
REFRESH primary command 993
REFS primary command 994
related
information, for ZDT/Db2 xv
table, edit 874, 887
tables, listing 193
Related tables panel, description 787
release, displaying ZDT/Db2 27
Remote Db2 Location Selection panel, description 791
remove trailing spaces option 669
REORG Utility (Indexes) panel, description 792
REORG Utility (Indexes) with LISTDEF panel, description 793
REORG Utility (Table Spaces) panel, description 795
REORG Utility (Table Spaces) with LISTDEF panel, description 797
REORG, generating Db2 utility 371
repeat
line (row selection criteria) 802
row 874, 887
row (partition information) 570
repeatable
read keep locks option 679
read option 678
repeatable items, syntax diagrams xv
repeating a search for a string 146
replace existing data, Db2 LOAD utility option 739
RESET primary command
description 994
in editor sessions 187
resume (append new data), Db2 LOAD utility option 739
Retrieve foreign key information when building templates processing option 862
retrieve row from clipboard 870, 884
return code, maximum when compiling a COBOL copybook 820
return code, maximum when compiling a PL/I copybook 827
RETURN instruction, return values
DROP 278
STOP 279
STOP IMMEDIATE 279
RETURN statement 268
reuse existing data, Db2 LOAD utility option 739
reverse engineering
description 324
GEN line command 324
list of objects 324
performance tips 326
Revoke Application Plan Privileges panel, description as a grant privileges panel 798
Revoke Buffer Pool Privileges panel, description as a grant privileges panel 798
Revoke Collection Privileges panel, description as a grant privileges panel 798
Revoke Column Privileges panel, description as a grant privileges panel 798
Revoke Database Privileges panel, description as a grant privileges panel 798
Revoke Distinct Type Privileges panel, description as a grant privileges panel 798
Revoke Function Privileges panel, description as a grant privileges panel 799
Revoke Package Privileges panel, description as a grant privileges panel 799
revoke privileges panels
description 798
list of 798
Revoke Schema Privileges panel, description as a grant privileges panel 799
Revoke Sequence Privileges panel, description as a grant privileges panel 799
Revoke Storage Group Privileges panel, description as a grant privileges panel 799
Revoke Stored Procedure Privileges panel, description as a grant privileges panel 799
Revoke System Privileges panel, description as a grant privileges panel 799
Revoke Table Privileges panel, description as a grant privileges panel 799
Revoke Table Space Use Privileges panel, description as a grant privileges panel 799
revoking
privileges 344
REXX
arithmetic operators 263
boolean operators 265
built-in functions 266
comparison operators 261

converting Db2 null indicator 301
expressions 258
functions, built-in 266
general information 258
logical operators 265
procedures, using with Import, description 268, 279
using procedure in Import (option 3.6) 257
variables
INREC 269
OUTREC 269
REXX, print option 829
RFIND primary command 995
RG row selection criteria operator 804
RIGHT primary command 996
Routine Parameters panel, description 799
row
convert to
lowercase 873, 886
uppercase 876, 889
copy 869, 883
count, option 613, 617, 633
delete 869, 883
delete (partition information) 571
display excluded 870, 884
exclude from display 876, 889
get from clipboard 870, 884
insert 871, 884
insert (partition information) 570
label 876, 889
labels 150
move 873, 887
overlaying 873
repeat 874, 887
repeat (partition information) 571
retrieve from clipboard 870, 884
save to clipboard 875, 888
ROW line command
application plans 475
collections 484
columns 505
databases 611
DBRMs 608
distinct types 639
functions 712
indexes 726
Object List (option 3.4) 322
packages 472
schemas 813
sequences 817
storage group storage group usage
privileges 852, 920
storage groups 850
stored procedures 854
synonyms 856
table spaces 881
tables, views and aliases 895
triggers 904
row selection criteria
connectors, list of 802
information stored in templates 75
operators, list of 803
prefix commands, list of 800
Row Selection Criteria panel, description 799
ROWID data type, description 442
rows
printing 331
scrolling to specified number 142
selecting
for display 86
to view in editor session 139
using to limit data displayed 150, 162
RP primary command 997
RR
edit prefix command 875, 888
line command, row selection criteria 802
rules, column mapping 118
running Z Data Tools/Db2
batch mode 432
RUNSTATS Utility (Indexes) panel, description 805
RUNSTATS Utility (Indexes) with LISTDEF panel, description 807
RUNSTATS Utility (Table Spaces) panel, description 808
RUNSTATS Utility (Table Spaces) with LISTDEF panel, description 810
RUNSTATS, generating Db2 utility 376
RUNTEMP primary command 998

S

S
edit prefix command 875, 888
line command
application plans 475
databases 611
packages 472
storage groups 850
tables, views and aliases 895
SAF-rule controlled auditing
Create audit trail option
Copy Utility (From) panel 516
Data Create Utility panel 606
Db2 Edit panel 619
Db2 View panel 634
Edit/Execute SQL (Data Set) panel 643
Editor Options (1 of 8) panel 646
Enter, Execute and Explain SQL panel 685
Export Utility panel 701
Import (To) panel 722
Print Utility panel 767
specifying Db2 object 137
description 71
sampling data
clustered 170
random 172
sampling frequency option 654
sampling seed option 654
sampling type option 651
SAVE primary command 999
save row to clipboard 875, 888
SAVEAS primary command 999
saving changes, editor session 127
SBCS data, Db2 LOAD utility option 739
SCH line command
distinct types 639
functions 712
stored procedures 854
triggers 904
Schema Privilege Details panel, description as a privilege details panel 768
Schema Privileges panel, description as a privileges panel 768
Schemas panel, description 812
scramble
options
range values, specifying 111
scramble type, specifying 109
value columns, specifying 110
value data set name, specifying 111
value list 112
value option, specifying 110
scrambling data
copy, import, export 292
scrollable cursor type option 651
scrolling 100, 160
in panels 36, 37
to first page 37
to labeled row 142
to last page 36
to specified column 142
to specified row 142
search
direction, LOCATE primary command 144
SEL line command
tables, views and aliases 896
Select Statement Browse panel, description 815
Select Statement Edit panel, description 815
selecting
column for attribute change 104
options on ZDT/Db2 panels 45
rows 139
selecting a column for specifying scrambling options 108
self-referencing constraint 195, 198
sensitive cursor option 651
sequence of data, changing 101
Sequences panel, description 816, 816
session, starting another 135
Set Batch Job Card Information panel, description 817
Set COBOL Processing Options panel, description 818
Set PL/I Processing Options panel, description 825
Set Print Processing Options panel, description 827
Set Processing Options
HLASM Processing Options 821
Set Processing Options panel, description 832
Set System Processing Options panel, description 835
Set Utility Processing Options panel, description 840
Settings (option 0), description 832
shadow lines
assigning labels to 877, 890
hiding or showing 1000
SHADOW primary command 1000
show
column number option
single display format 659, 660
table display format 656
data type option 656
end of string option 668
foreign key option 664
index indicators option 665
keys, indexes 659
primary key option 663
scale line option 656
table display format 656
show excluded shadow lines option 661
Show export options panel option 666
show prefix area on the right option 658
SHOW primary command 34, 1001, 1001
show SQLCODE option 658
SHOWCOB primary command
description 1003
using 29
showing
leading zeros 107

SHOWPLI primary command
description 1003
single display format
description 156
options 659
skip count option 653
SMALLINT data type, description 441
Sn edit prefix command 875, 888
SNGL display format , *See* single display format
Sort Fields panel, description 841
SORT primary command
application plans 475
collections 484
columns 505
databases 612
DBRMs 608
distinct types 639
functions 712
indexes 726
packages 472
schemas 814
stored procedures 854
synonyms 856
table spaces 882
tables, views, and aliases 896
triggers 905
view or edit 1003
sorting
data 164
displayed data (Object List) 323
SQL
errors 427
keywords 210
line command
application plans 475
collections 484
packages 472
triggers 904
optimization options 169
primary command 1005
prototyping, execution and analysis (option 4) 390
prototyping, execution and analysis (option 4), description 390
Prototyping, Execution and Analysis panel, description 843
statements
working with 390
SQL prototyping
basic 390
SQLID
primary command 1006
specifying 52
SR line command
EXPLAIN plan table 757
plan table rows panel 759
SS line command
table edit 875, 888
SSID command, changing Db2 subsystem 32
Start Db2 Interactive (option 5), description 438
start position for editor session, specifying 126
starting
point, LOCATE primary command 144
ZDT/Db2 editor session 123
starting another session
ZDT/Db2 editor 135
statement continuation 432
Statement Table Rows panel, description 845
STOP IMMEDIATE, RETURN return value 279
STOP, RETURN return value 279
Storage Group Usage Privileges panel, description 851
Storage Group(s) panel, description 848, 848
Stored Procedure Privilege Details panel, description as a privilege details panel 768
Stored Procedure Privileges panel, description as a privileges panel 768
Stored Procedures panel, description 852
subsystem, Db2
selecting 30
Subsystem Selection menu 33
summary statistics 337
suppression of leading zeros 107
SYN line command
tables, views and aliases 896
Synonym Dependencies panel, description 855
Synonym Details panel, description as a details panel 636
synonyms
panel, description 855
using when specifying an object 47, 49, 49, 50, 50
syntax diagrams, how to read xv
SYSLIB data sets
COBOL processing option 819
PL/I processing option 826
SYSPRINT
REXX procedures, TALLY external function 276
SYSPRINT, print option 828
system options 64, 65

T

T line command
application plans 475
columns 505
databases 611
distinct types 639
EXPLAIN plan table 757
indexes 726
plan table rows panel 759
synonyms 856
table spaces 881
tables, views and aliases 896
triggers 904
TABL display format , *See* table display format
table
display format options 656
Table Check Constraint Dependencies panel, description 868
Table Check Constraint Details panel, description 868
Table Check Constraints panel, description 868
Table Details panel, description as a details panel 636
table display format 153
Table Privileges panel, description as a privileges panel 768
Table Selection panel, description 878
table space
for a Db2 object, specifying 49
Table Space Details panel, description as a details panel 636
Table Space Parts panel, description 879
Table Space Use Privilege Details panel, description as a privilege details panel 768
Table Space Use Privileges panel, description as a privileges panel 768
Table Spaces panel, description 879
Table/Single format, prefix area and shadow options 647
Table/View Privilege Details panel, description as a privilege details panel 768
Table/View/Alias Selection panel, description 890
tables
large, options for 170
Tables, Views and Aliases panel, description 891
tabular display format , *See* table display format
TALLY external function 276
TB line command 472
TE , *See* TEDIT primary command
TEDIT primary command 84, 152, 1006
template
creating copybook from 121
Template Mapping panel, description 900
TEMPLATE Options (1 of 2) panel, description 897
TEMPLATE Options (2 of 2) panel, description 898
Template Save panel, description 902
template utility
options 70
templates
adding a description 85
COBOL data description entries 78
definition 21
editing
changing column headings 105
changing the sequence of displayed data 101
description 84
holding columns for scrolling 100
HLASM data description entries 78
how to use 74
information stored in 75
mapping
automatically by ZDT/Db2 113
data 112
rules (data type conversion) 118
specifying 114
where you can use 113
overview 21
PL/I DECLARE statements 79
printing 77
selecting
column for attribute change 104
columns for display 99
rows for display 86
using with non-Db2 data 77
where you can use 22
working with 74
temporary data set allocation, option 66
Terminal, print option 828
TIME data type, description 442
TM external function 277
TOP primary command 37, 1007
TP primary command 1007
TR line command
schemas 813
trace options
settings 67
Translate Db2 object names processing option 857

Translate input SQL statements processing option 858
Translate non-printable chars, print option 831
Trigger Details panel, description as a details panel 637
Triggers panel, description 903
TSO region size
  minimum supported 25
  storage considerations 138
  viewing large tables or views 139
tutorial help 57
TYPE primary command 1008
type, export options 693
types of Db2 data 440

U

UC edit prefix command 875, 888
UCC line command 876, 889
UH line command (for EXPLAIN plan table) 758
underscore (_)
  filtering object lists 309
  using in ZDT/Db2 entry fields 50
UNDO primary command 1008
Unicode data 179
unique constraints, create table option 232
UNLOAD Options panel, description 905
unload utility
  options 70
UNLOAD Utility (Table Spaces) panel, description 908
UNLOAD Utility (Table Spaces) with LISTDEF panel, description 909
UNLOAD Utility (Tables) "From" panel, description 910
UNLOAD Utility (Tables) "To" panel, description 911
UNLOAD Utility (Tables) with LISTDEF panel, description 914
UNLOAD, generating Db2 utility 380
UP primary command 37
Update
  materialized query tables 191
Uppercase message text, print option 831
uppercase, converting data to 189
usage, export options 692
Use IEEE floating point, Db2 LOAD utility option 738
Use row count value option 650
Use uncommitted read
  copy option 510
Use uncommitted read when accessing the Db2 catalog processing option 861
Use/Keep exclusive locks option 679
Use/Keep share locks option 679
Use/Keep update locks option 679
using a template 22
utilities 68
Utilities (option 3.9)
  description 346
  generating a Db2 utility job 350
  setting options 348
Utilities menu 447
Utility Functions menu
  Audit trail (option 3.10) 335
  Copy (option 3.3) 246
  Export (option 3.7) 279, 287
  Import (option 3.6) 251, 279
  Object List (option 3.4) 303
  Objects (option 3.2) 208
  Print (option 3.1) 328, 331
  Print Browse (option 3.11) 333, 990
  Privileges (option 3.5) 339
Utility Functions panel, description 915
UTL line command
  indexes 726
  table spaces 881
  tables, views and aliases 896

V

V line command
  application plans 475
  packages 472
  tables, views and aliases 896
valid commands (Object List Utility) 320
validating changes, editor session 127
value list, scrambling data 112
VARBINARY data type, description 440
VARCHAR
  data type
    description 440
    processing options 178
  options for processing 178
VARGRAPHIC data type, description 440
variables, syntax diagrams xv
VARYING 82
Varying length and nullable column options 648
varying-length columns, options for processing 178
VARYINGZ 82
VER
  line command
    packages 472
  primary command 27, 1010
View (option 1)
  displaying data in hexadecimal format 165
  limiting data displayed
    excluding rows 162
  scrolling 160
  sorting data 164
  VARCHAR, options for processing 178
  varying-length columns, options for processing 178
  zooming 158
View Details panel, description as a details panel 636
View options 67
VIEW primary command 1010
VOL line command
  storage groups 850
Volumes panel, description 919
VS line command
  tables, views and aliases 896

W

Wide print, print option 830
WIDTH primary command 1011
wildcards, using in ZDT/Db2 entry fields 50
working with object list panels 307
WRITE external function 277

X

X
  edit prefix command 876, 889
  line command
    application plans 475
    columns 505
    databases 611
    EXPLAIN plan table 758
    packages 472
    storage groups 850
    tables, views and aliases 896
XC line command
  tables, views and aliases 896
XMLBRWS primary command 1013
XMLEDIT primary command 1014
XMLLEN primary command 1015
XMLVIEW primary command 1015
XPT primary command
  storage groups 850
XX line command 876, 889

Z

ZCSR editor-assigned label 151
ZDT/Db2
  Db2 releases supported by 24
  entry fields
    patterns, using 50
    wildcards, using 50
  exiting 29
  external functions
    CHANGE 271
    CONTAINS 272
    FLD 273
    NCONTAIN 274
    PRINT 274
    PRTCOUNT 275
    RECSIN 275
    RECSOUT 275
    TALLY 276
    TM 277
    WRITE 277
  flavors xiii
  functions
    BATSQL (Batch SQL) 1019
    D2G (Data Generate) 1058
    D2TP (Template create/update) 1060
    DBC (Copy) 1027
    DBI (Import) 1034
    DBP (Print) 1040
    DBX (Export) 1043
    description 1018
  functions, overview 21
  launching 26
  Object List Options (1 of 2) panel, description 743
  Object List Options (2 of 2) panel, description 746
  panels
    action bar 38
    command line 39
    database, specifying 49
    Db2 object object names, specifying 46
    function keys 39
    location, specifying 47
    name of a Db2 object, specifying 49
    owner, specifying 48
    panel body 39
    panel title 39
    record count 39
    selecting options 45
    SQLID, specifying 52
    table space, specifying 50
    working with 38
  REXX procedures, interfacing with, description 268, 279
  Systems Options (1 of 4) panel, description 857
  Systems Options (2 of 4) panel, description 859
  Systems Options (3 of 4) panel, description 862

Systems Options (4 of 4) panel,
description 864
template, creating copybook from 121
ZDT/Db2 editor
ending a session 133
setting options for the current ZDT/Db2
session 126
specifying a Db2 object 136
specifying the type of session 124
starting a session 123
starting another session 135
validating, saving changes 127
ZDT/Db2 Release News panel, description 708
ZDT/Db2 session, starting another 135
ZFST editor-assigned label 151
ZLST editor-assigned label 151
Zoom function key 160
ZOOM primary command 160, 1016
zooming 158