

HCL OneDB 2.0.1

Installing



Contents

- Chapter 1. Installing..... 3**
 - HCL OneDB™ server..... 3
 - Installing HCL OneDB™..... 3
 - Client products..... 34
 - HCL OneDB™ Client SDK..... 34
- Index..... 58**

Chapter 1. Installing

These topics describe how to install HCL OneDB™ database servers, client products, and DataBlade® modules.

Installing HCL OneDB™ database servers

These topics describe how to install HCL OneDB™ database servers.

Installing HCL OneDB™

The *HCL OneDB™ Installation Guide* describes how to install, configure, and initialize the HCL OneDB™ database server.

Typically, database administrators (DBAs) install the database server. The documentation assumes that you are familiar with the operating procedures of your computer and with your operating system.

Preparing for installation

Before you install HCL OneDB™ database server products, you must prepare your system and plan for choices that you must make during the installation process.

About this task

Preparing for installation includes reviewing system requirements, planning how to upgrade, deciding on the installation owner and the installation directory, reviewing installation options, and gathering information if you want to create a database server during installation. It also includes preinstallation tasks such as obtaining the software and reviewing the machine notes.

To prepare for installation:

1. Obtain the software and configure a License Server to obtain licenses to run HCL OneDB™.
You can download HCL OneDB™ software from [HCL License and Delivery portal](#)
For more information about HCL OneDB™ see, <https://www.hcltechsw.com/products/onedb>.
2. Review the HCL OneDB™ components that can be installed with the database server.
You can install all of them or some of them.
3. Ensure that your computer meets the [system requirements on page 5](#).
4. Choose a user account with which to install the product. The user account becomes the installation owner.
 - UNIX, Linux: Decide between a non-root owner and an owner with root or administrative privileges.
 - If you plan to install as an administrative user, decide on a password for user **informix**.
5. Choose an installation directory.
If the database server is already installed, decide on the upgrade path.
6. Decide between a typical or custom installation.
 - a. Choose a typical installation in the following cases:
 - For default server features, language support, extensions, and tools.
 - To create a configured database during installation with connectivity for JSON applications and multiple storage spaces for data.
 - b. Choose a custom installation if you want to have any of the following options:

- To install a subset of database server components.
- In root-based installation, to set up event auditing for the security policies of your organization.
- To create a configured database server during installation with one storage space for data.
- To prepare a response file for a non-interactive installation.

Depending on your operating system, you can choose other types of installations that have similar options to a custom installation. UNIX, Linux: You can choose the Legacy installation to prepare a script-based installation. Linux: You can choose the RPM installation to generate an RPM package.

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

What to do next

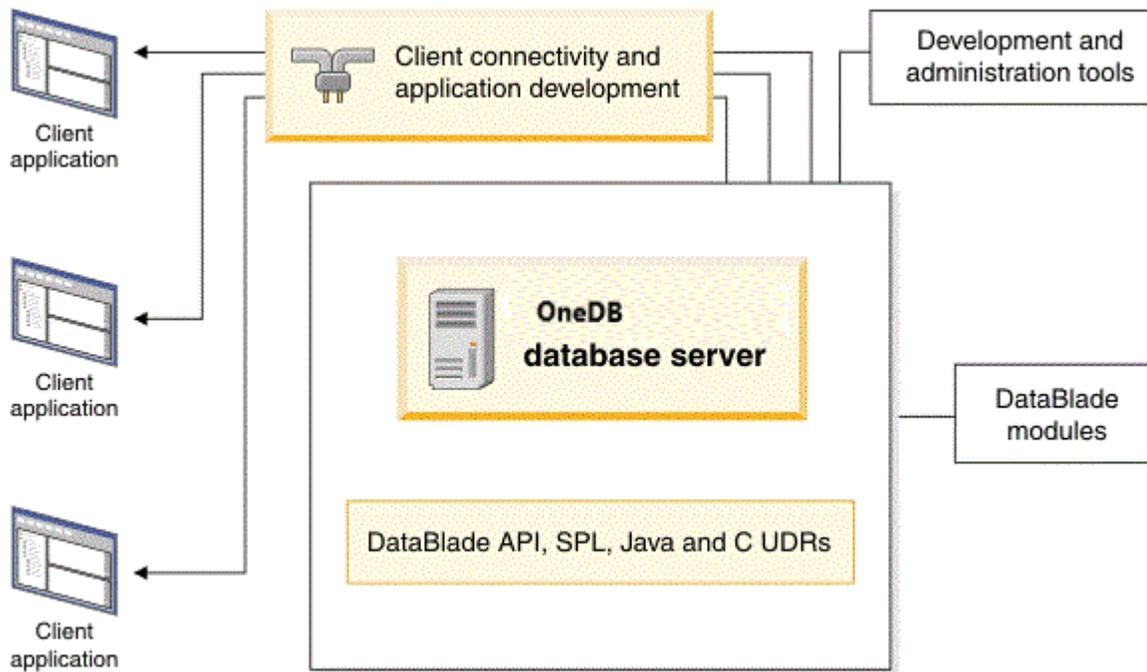
After you install the database server, you can create the demonstration databases that are included with the database server. Many examples in the documentation and in the `$ONEDB_HOME/demo` (UNIX, Linux) or the `%ONEDB_HOME%\demo` (Windows) directory are based on the demonstration databases. You can use the demonstration databases to explore database server features. You can create the demonstration databases at any time after installation by running the `dbaccessdemo` command. See [Demonstration databases on page](#) .

HCL OneDB™ components

When you install the HCL OneDB™ database servers, you can also install related products, such as client APIs.

The following diagram provides an overview of the base architecture of the HCL OneDB™ database server and related products.

Figure 1. Architecture of HCL OneDB™ database server and related products



Depending on your edition of HCL OneDB™ and your operating system, the product package includes installation programs for some or all of the following associated products:

HCL OneDB™ Client Software Development Kit

Contains client APIs for developing and running client applications.

HCL OneDB™ JDBC Driver

Java™ interfaces and classes to connect to HCL OneDB™ databases.

System requirements for HCL OneDB™ database server

This section provides detailed information about the system requirements for HCL OneDB™ database server.

Hardware Requirements

Before you install the HCL OneDB™ database server, make sure that your computer meets the minimum hardware requirements.

Disk space and memory requirements for installation

The following disk space and memory requirements for installing the database server are approximate. The requirements for some operating systems might be lower.

Disk space

Approximately 750 MB of disk space is required for a typical installation.

Some installation choices require more disk space. The installation application informs you of the total disk space that is required by your setup before you copy the binary files to your host computer.

RAM

Approximately 1 GB RAM is required.

Temporary disk space

You can change the location of the temporary directory by setting the **IATEMPDIR** environment variable before you run the installation application.

UNIX™, Linux: 2 GB free space is required in your temporary directory (`/tmp` by default). If the temporary directory does not have enough space, the installation application attempts to write the information into the home directory of the user account.

Windows™: 1 GB free space is required in your temporary directory (`C:\tmp` by default).

Software Requirements

Before you install the HCL OneDB™ server, make sure that your computer meets these software requirements.

Operating system requirements

Install the required operating system patches and library files that are listed in the machine notes for your operating system in the doc directory of the media files before installation.

UNIX™, Linux™: The directory on which you plan to install the database server must be local or an NFS-mounted file system with regular operating-system files.

Windows™: The drive on which you plan to install the database server must be formatted with NTFS and must have 8.3 file name creation enabled. Turn off User Account Control (UAC) security before you install the database server.

HCL OneDB™ Client SDK and Server 1.0 is certified to run on 64-bit versions of these operating systems:

Table 1.

Operating System	Version
Linux x86-64	RHEL 7.4 / CentOS 7.4 <ul style="list-style-type: none"> • Base kernel version 3.10.0 • Glibc version 2.17 • Compiler 4.8.5 RHEL 8

Table 1. (continued)

Operating System	Version
	<ul style="list-style-type: none"> • Base kernel version 4.18.0 • Glibc version 2.28 • Compiler 8.2.1 <p>SuSE SLES 12</p> <ul style="list-style-type: none"> • Minimum service pack 3 • Base kernel version 4.4.73 • Glibc version 2.22 • Compiler 4.8.5 <p>SuSE SLES 15</p> <ul style="list-style-type: none"> • Base kernel version 4.12.14 • Glibc version 2.26 • Compiler 7.3.1 <p>Ubuntu 16.04 LTS</p> <ul style="list-style-type: none"> • Base kernel version 4.4.0 • Glibc version 2.23 • Compiler 5.4.0 <p>Ubuntu 18.04 LTS</p> <ul style="list-style-type: none"> • Base kernel version 4.15 • Glibc version 2.27 • Compiler 7.5 <p>Ubuntu 20.04 LTS</p> <ul style="list-style-type: none"> • Base kernel version 5.4 • Glibc version 2.31 • Compiler 9.3
AIX	7.2
Windows	<p>Windows 2016</p> <p>Windows 2019</p> <p>Windows 10</p>

Application software required to run/install Server

Table 2.

Operating System	Version
Linux	OpenSSL any of the supported versions 1.0.2, 1.1.0 or 1.1.1.
AIX	OpenSSL 1.0.2 is required
Java8	JRE is required to install OneDB. Some components of OneDB also require Java8 Windows install requires a 1.8.0 JRE Linux/AIX install requires a 1.8.0 or later JRE

Installation owner

The owner of the OneDB database server depends on the privileges of the user who runs the installation application. The owner of the server creates and controls all other user accounts that can access the database server.

The standard way to install the database server is as a superuser with administrative privileges. The installation application creates the user **informix**. User **informix** is a user account with main authority over the database server instance.

UNIX, Linux: If you install the database server as a non-administrative, or non-root, user, you become the owner of the database server. In this case, user **informix** is not created. You cannot change the owner of the database server. You cannot convert or upgrade a non-root installation to a standard, root-based installation, and vice versa.

User **informix**

User **informix** is required for root-based installations because it has the unique user identifier (UID) to manage and maintain database server instances and databases on the host server.

UNIX™, Linux: User **informix** is a member of the group **informix**. On UNIX™ or Linux™, if group **informix** exists on your system, but user **informix** does not exist, you must create user **informix** before you install the database server.

Windows™: User **informix** is a member of the Informix-Admin group and the Administrators group. In most cases, the installation application automatically creates the user **informix** and the group **informix** or Informix-Admin group. User **informix** can log on as a service and act as part of the operating system.

If you are installing the database server for the first time on your system, the installation application prompts you to provide a password for the **informix** user. If user **informix** exists on your system, the installation application prompts you to confirm the password.

The password for the **informix** user account must be protected. Let only trusted database and security administrators log in as user **informix**.



Important: The database server does not start if password standards for user **informix** or any other users do not conform to local security standards.

UNIX, Linux: Non-administrative, or non-root owner

If you run the installation application as a non-administrative user, that user is the *non-root owner* of the installation and has database server administrator (DBSA) privileges over the database server.

The installation directory must be on a local file system if you plan to use onipcstr connections because you cannot create a named domain socket on a non-local file system.

A non-root installation is appropriate in the following situations:

- You plan to embed the database server in an application that is to be deployed on other computers where you either do not want a root-level installation or you want non-administrative users to install the application.
- You do not have or do not want to use root user credentials for the HCL OneDB™ database server installation.
- You are developing a virtual appliance.

If you install the database server without root privileges, you cannot use the following features and tools:

- Enterprise Replication for servers that have different owners
- High-availability clusters
- Auditing through role separation
- The ON-Bar utility for backing up and restoring data

Encrypted connections and column-level encryption (CLE) are supported only if the required security-layer plug-in is installed separately on the host computer.

Creating the group informix and user informix (UNIX, Linux)

In a few situations during a root-based installation, you must create the group **informix** and user **informix** before you install the database server on UNIX or Linux operating systems.

Before you begin

You must have root authority to create users or groups.

About this task

You must create the objects before you start the installation application in the following situations:

- You want to specify a particular identifier (ID) number.
- The group **informix** exists on the system; however, the user **informix** does not. In this case, you must create only user **informix**.

To create the objects:

1. If necessary, create the group **informix** by using the following command, where *n* is a unique identifier (ID) greater than 100:

Choose from:

- AIX: `mkgroup -a n informix`
- UNIX, Linux: `groupadd -g n informix`

2. Create the user **informix** by using the `useradd` command, where *n* is a unique identifier (ID) greater than 100:

Choose from:

- `useradd -u n -g informix informix`

Result



Important: Add users to the group **informix** only if the users need administrative access to the database server.

3. Create a password for user **informix** by using the `passwd` command.

Installation directory

You can use the default installation directory for the database server when you run the installation application, or you can create a directory before you install and then select that directory during installation.

The installation directory is known as the `ONEDB_HOME` directory because the environment variable **ONEDB_HOME** is set to the installation directory.

The directory where you install HCL OneDB™ products must fulfill these requirements:

- The directory must be empty, except in the following situations:
 - You are upgrading to a fix pack of the same version.
 - You accept the risks of upgrading to a new version by replacing the existing version.
- The full path to the installation directory must not exceed 200 characters, including path separators, and must not contain spaces. UNIX™, Linux: The directory for a non-root installation must not exceed 60 characters.
- The installation path, including all path elements, must be secure before the database server can start.



Important: The database server will not start unless the installation directory is secure.

UNIX™, Linux™: If the installation application finds a security issue with the installation directory, you must choose one of the following methods to secure the directory:

Let the installation program secure the path (recommended)

The installation continues and generated script fixes nonsecure directories along the installation path.

Continue installation; manually run script to secure the path later

After installation is complete, you must run the `onsecurity` utility to generate a security script, and then run the security script. The `onsecurity` utility is in the `/SERVER/` directory on the installation media. After installation, you can find the utility in the `$ONEDB_HOME/bin` directory.

View other options to make the installation directory secure

You must select how to reset the permissions for the owner, group, and installation directory. You must understand what constitutes a secure installation path in your environment, including the requirements of user **informix** and group **informix**.

Installation options

The primary way to install the HCL OneDB™ database server is with an interactive installation that prompts you to configure installation properties. After you run an interactive installation, you can run non-interactive installations that are based on the original installation.

Interactive database server installation setup options

An interactive database server installation has the following setup options:

Typical installation

Recommended. You install the database server with all associated components, HCL OneDB™ Client Software Development Kit, and the HCL OneDB™ JDBC Driver.

Custom installation

You can minimize the size of the database server on disk by excluding unneeded features and products.

You can choose which client products to install. To determine whether you need to install client products, see *HCL OneDB™ Client Products Installation Guide*.

You can set up auditing through role separation.

You can prepare for a silent installation by creating a response file.

UNIX, Linux: Legacy installation

You have the same options as a custom installation, plus the product files are extracted to create a script-based non-interactive installation.

Linux: RPM installation

You have the same options as a custom installation, plus an RPM image is generated to create an RPM non-interactive installation.

For all setup options, you have the following choices:

- Whether to create a configured database server during installation. The database server that is created during installation differs slightly for a typical or custom installation. If you do not create a database server during installation, you must manually configure and start the database server after installation.
- The location of the installation directory.
- The identity of the installation owner.

Non-interactive installation methods

Non-interactive installation methods, such as a silent installation or a script-based installation, are based on the results of interactive installation or on an existing database server installation.

Event auditing

If you choose a custom setup, you can enable event auditing. Event auditing tracks selected activities that users perform. You can improve the security of your event-auditing procedures by configuring role separation, which provides members of certain group identifiers (group IDs) on your system the privileges to manage and examine auditing records. Role separation provides increased database security because the database server separates administrative tasks into mutually exclusive roles.



Important:

- Role separation is not supported in a non-root installation.
- You must select custom installation setup to enable role separation.
- If you enable role separation, you cannot turn it off after the database server is installed. To remove role separation, you must uninstall the database server and reinstall it without role separation.

UNIX™, Linux™: Role separation

If you do not enable role separation, the **informix** group has privileges to perform all administrative tasks.

Role separation provides two roles:

Database System Security Officer (DBSSO)

Controls what the auditing subsystem monitors and which actions database users can perform.

Auditing Analysis Officer (AAO)

Controls whether auditing occurs, maintains the audit log files, and analyzes the audit records.

The **informix** group is the default group that is associated with the two roles. During installation, you can replace the default groups with existing groups.

After installation is complete, establish an audit-only user account for each individual who acts as a DBSSO or AAO. For example, a person with DBSSO responsibilities can have the user **DBSSO1** account, and also have the user **garcia5** account for general database server access.

Windows™: Role separation

If you do not enable role separation, the **Informix®-Admin** group has privileges to perform all administrative tasks.

If you enable role separation during installation, you are prompted to create groups and users and add the users to the corresponding groups. During installation, you can replace the default users and groups with existing users or groups.

Table 3. Role separation

Header	Header	Header
Information Administration	General Database Administration	Performs general administrative tasks, such as archiving and restoring data, monitoring use and performance, and tuning the system.
ix_db_sso	Database Security Officer	Maintains the security of the database server. Functions of this role include audit adjustment and changing security characteristics of storage objects. Creation of this user role requires selection of a password during installation.
ix_aao	Auditing Analysis Officer	Audits the records of specific types of database activities. If someone attempts to circumvent or corrupt the security mechanism of the database, these actions can be traced. Creation of this user role requires selection of a password during installation.
ix_users	Database Users	Accesses the database to perform user tasks. Only users who are designated as members of the ix_users group can access the database.

Exclude database server components

If you choose a custom setup, you can minimize the size of the database server on disk by excluding some components. Reducing the size of the installation can be useful if you plan to embed the database server in other systems or applications.

The HCL OneDB™ database server consists of discrete, separately installable components. You can select to install only the database server components that your application and deployment require. For example, you might not need replication capabilities. Some components are mutually dependent, and must coexist in the instance. The installation application enforces these dependencies by automatically including dependent components or informing you when a combination of selections is not supported.

You can easily modify your installation by adding or removing separately installable components after the database server is installed without installing the base database server again. Adding or removing components after you installed the database server does not affect the integrity of your system. However, all HCL OneDB™ components must run on the same version as the core database server.

Database server configuration during installation

You can choose to create and configure a database server during installation. The database server is configured by the choices that you make in the installation application. After installation is complete, disk space for the database server is initialized and the database server is started.

All mandatory configuration parameters and environment variables are set and default connectivity information is configured. The database server is also configured to require minimal administration. The database server is configured to automatically tune resources for performance, which include expanding the buffer pool, the logical logs, and the physical log, and increasing virtual processors. You can further customize the database server by setting other configuration parameters and environment variables.

 **Important:** If you plan to use a locale or language other than the default (US English), set the appropriate global language support (GLS) environment variables before you create a database. For example, to prevent server and collation problems, it is crucial to set the `GL_USEGLU` environment variable correctly. The same Unicode collation must be used for the database and the client application environment.

The following storage spaces are created for an installation of the database server:

- A root dbspace to contain control and tracking information
- An extendable plogspace for the physical log
- A dbspace for the logical logs
- A temporary dbspace
- An sbspace
- A temporary sbspace
- Dbspaces for data:
 - Typical installation: Three extendable dbspaces with the default page size, and three extendable dbspaces with an 8 KB page size
 - Custom installation: One dbspace with the default page size

The sizes of the spaces are based on the number of expected users that you specify during the installation. The spaces are not encrypted. You can encrypt the storage spaces that are created during installation by enabling storage space encryption, running a backup, and then running a restore that encrypts the spaces.

If you run a typical installation, the server is configured with a JSON wire listener for connections to MongoDB applications. The JSON wire listener is started, and connected to the database server through the **ifxjson** user.

Windows: A program group for the database server is created from which you can open a command prompt for the server, start the Server Instance Manager, uninstall the database server, or view documentation.

Running an interactive installation

You can install the HCL OneDB™ database server by running the interactive installation application. You can choose a typical or a custom setup.

Before you begin



Important: If you want to create a response file to use as a template for installing the database server on other computers, follow the procedure in [Preparing a response file on page 16](#).

Before you install the database server, complete the following tasks:

- [Prepare your system for installation on page 3](#).
- Log in as the appropriate user to be the owner of the database server.

About this task

You can include options to the installation command if you run the command from the command line. For example, you can change the name or location of the installation log file.

To install HCL OneDB™:

1. Start the installation application from the media directory.

Choose from:

- UNIX™, Linux™:
 - a. If necessary, extract the product files. For example, run the tar command:

```
tar xvf filename
```

The *filename* is the name of the product `tar` file.

- b. Run the following command: `onedb_install`

- Windows™: Double-click the `onedb_install.exe` file.

2. Select the setup method, for example, **Typical** or **Custom**, and follow the instructions in the installation application.
3. Complete the installation and exit the installation application.

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

If you created a database server during installation, the database server is configured, initialized, started, and ready for client connections. To see the connectivity information, look at the `sqlhosts` file. To see the configuration information, look at the `onconfig` file. Both files are in the `$ONEDB_HOME/etc` (UNIX, Linux) or the `%ONEDB_HOME%\etc` (Windows) directory. You can customize the database server properties by setting configuration parameters in the `onconfig` file, setting environment variables in your environment, and adding connectivity information to the `sqlhosts` file. See [Overview of database server configuration and administration on page](#) .

What to do next

If you did not create a database server during installation, you must configure and initialize the database server.

Running non-interactive installations

You can run non-interactive installations after you first run an interactive installation during which you create an installation template.

A silent, or unattended, installation requires that you create a response file, copy the product installation files and the response file to the target computer, and then run the silent installation command. The response file acts as a template for the silent installation.

UNIX, Linux: A scripted installation requires that you copy an existing product installation directory to the target computer and run a script.

If you use Enterprise Replication or high-availability clusters, you can create a server clone from a snapshot of an existing database server with the `ifxclone` utility.

If you plan to embed HCL OneDB™ products in your application, you can use one of the following methods to deploy the database server:

- You can use the deployment utility to deploy snapshots of preconfigured database servers on one or more computers. A snapshot is an image of a database server that includes the installation directory, configuration settings, and any data spaces that are associated with the instance. The installation can be a working instance, or an installation that you set up as a template from which to deploy the instance on other computers. You can use the deployment assistant to customize the snapshot.
- Linux: You can run an RPM installation to create an RPM image of a customized database server installation, and then deploy the image through a silent installation.

Preparing a response file

Before you run a silent installation, you must run an interactive installation to create a response file. The response file contains information about how to install the database server on other computers that have the same basic type of operating system: either a UNIX operating system (including Linux), or a Windows operating system.

Before you begin

Before you start the interactive installation, complete the following tasks:

- [Prepare your system for installation on page 3.](#)
- Log in as the appropriate user to be the owner of the database server.

About this task

The response file contains installation settings for a product and its features. You have the following choices for the response file:

Default response file

The default response file, `bundle.properties`, produces a typical installation. The `bundle.properties` file is in the top directory of the installation media. You must either edit the `bundle.properties` file to set the `LICENSE_ACCEPTED` property to `TRUE`, or include the `-DLICENSE_ACCEPTED=TRUE` option in the silent installation command.

Response file that is generated by an interactive installation

A generated response file contains the installation settings of a custom interactive installation. Use the response file to run a silent installation on another computer.

If you chose to create a database server during this installation, the database server is also created and configured when you run the silent installation.

To generate a custom response file:

Run the installation command with the `-r` option and specify a path and name for the response file:

- UNIX™, Linux™:
 - a. If necessary, extract the product files by running the following command, where *filename* is the name of the product tar file: `tar xvf filename`
 - b. Run the following command: `onedb_install -r response_file_path`
- Windows™: Run the following command from a command prompt: `onedb_install.exe -i gui -r response_file_path`

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

If you created a database server during installation, the database server is configured, initialized, started, and ready for client connections. To see the connectivity information, look at the `sqlhosts` file. To see the configuration information, look at the `onconfig` file. Both files are in the `$ONEDB_HOME/etc` (UNIX, Linux) or the `%ONEDB_HOME%\etc` (Windows) directory. You can customize the database server properties by setting configuration parameters in the `onconfig` file, setting environment variables in your environment, and adding connectivity information to the `sqlhosts` file. See [Overview of database server configuration and administration on page](#) .

What to do next

After the custom response file is generated, you can use it to run a silent installation on another computer.

If you did not create a database server during installation, you must configure and initialize the database server.

Running a silent installation

You can run a silent installation command that does not require interaction.

Before you begin

Before you start a silent installation, complete the following tasks:

- [Prepare your target systems for installation on page 3](#).
- Log in as the appropriate user to be the owner of the database server.
- [Prepare a response file on page 16](#). You need different response files for UNIX-style operating systems and Windows operating systems.

About this task

On each target computer, you can edit the response file to customize the silent installation in the following ways:

- Accept the license. Include the LICENSE_ACCEPTED=TRUE property.
- Bypass operating system prerequisite checks. Use with caution. Include the PRQCHECK=FALSE property.
- Set the installation path. Include the USER_INSTALL_DIR=*path* property.
- Overwrite existing product installations. Use with caution. Include the OVERWRITE_PRODUCT=TRUE property.
- Create and start a database server during installation. Include the IDS_INSTANCE_CREATE=1 and IDS_INSTANCE_INIT=1 properties.

To run a silent installation on each target computer:

1. Copy the response file and the installation media to the computer where you want to install the database server.
2. Run the silent installation command, specifying the absolute path to the response file after the `-f` option:

Choose from:

- UNIX, Linux: `onedb_install -i silent -f response_file_path`
- Windows: `onedb_install.exe -i silent -f response_file_path`

You can include some of the silent installation properties in the command instead of setting them in the response file.

3. Complete any applicable postinstallation tasks, including initiating the disk space when you start the server for the first time.

If you choose to create and start a database server during installation, the database server is configured, initialized, started, and ready for client connections.

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

Example

Example

The following silent installation command specifies the default response file in the *media_dir* directory, specifies an installation directory of `opt/IBM/Informix` with the `-DUSER_INSTALL_DIR` property, and accepts the license with the `-DLICENSE_ACCEPTED=TRUE` property:

```
onedb_install -i silent -f media_dir/bundle.properties
-DUSER_INSTALL_DIR=opt/IBM/Informix -DLICENSE_ACCEPTED=TRUE
```

Installation command syntax

You can include options to the installation command when you run the command from the command line.

Log in as the appropriate user to be the owner of the database server to run this command.

Syntax

```
(1)
>>+-onedb_install-----+-- -i----->
```

```

| | (2) |
'-onedb_install.exe-----'

(1)
.-console-----
>----->
| | (1) | '- -r--response_file_path-' | (1) | |
| '-gui-----' | '- -DLEGACY=TRUE-----' |
'-silent-- -f--response_file_path-----'
'| Silent installation options |-'

>----->
'- -DDEBUG_LEVEL=level-'

>----->
'- -DDEBUG_FILE=debug_file_path-'

>-----<
'- -DLOG_FILE=log_file_path-' '- -help-'

```

Silent installation options

```

|----->
'- -DLICENSE_ACCEPTED=TRUE-'

>----->
'- -DOVERWRITE_PRODUCT=TRUE-' '- -DPRQCHECK=FALSE-'

>-----|
'- -DUSER_INSTALL_DIR=---path-'

```

Parameters

You can set properties on the installation command line that are useful for specific environments or installation methods. The values TRUE and FALSE are case-sensitive.

The following table describes the installation application options.

Table 4. Installation options

Option	Meaning
<code>-DDEBUG_FILE=debug_file_path</code>	Specifies the full path and name of the debug file. If tracing is enabled, the default name and location of the file is <code>/tmp/iad_dbg.log</code> (UNIX™, Linux™) or <code>\tmp\iad_dbg.log</code> (Windows™).
<code>-DDEBUG_LEVEL=level</code>	Specifies the level of tracing detail in the debug file. Possible values for <code>level</code> are 1 - 9, with 9 being the greatest level of tracing detail. By default, tracing is disabled.
<code>-DLEGACY=TRUE</code>	UNIX™, Linux™: Extracts the product files from the installation media to create a script-based redistribution to other computers. For instructions, see <i>Redistributing Informix with scripts</i> (UNIX, Linux).

Table 4. Installation options (continued)

Option	Meaning
-DLICENSE_ACCEPTED=TRUE	Accepts the product license terms during a silent installation instead of indicating acceptance in the response file.
-DLOG_FILE= <i>log_file_path</i>	Specifies the full path and name of the installation log file. The default name and location of the installation log file is <code>/tmp/iad_act.log</code> (UNIX™, Linux™) or <code>\tmp\iad_act.log</code> (Windows™).
-DOVERWRITE_PRODUCT=TRUE	Specifies to overwrite an existing product installation during a silent installation. Otherwise, the installation application exits if it finds an existing product installation.
-DPRQCHECK=FALSE	<p>Specifies to bypass the operating system prerequisites check during a silent installation.</p> <p> Important: Use with caution. Disabling the prerequisites check might result in an installation that does not function properly.</p>
-DUSER_INSTALL_DIR= <i>install_path</i>	Specifies the installation path during a silent installation instead of specifying the path in a response file. Replace <i>install_path</i> with the absolute path to the installation directory.
-help	Displays list of supported options and their functions.
-i gui	Starts the installation program in GUI mode.
-i console	UNIX™, Linux™: Starts the installation program in console mode.
-i silent -f <i>response_file_path</i>	<p>Starts a silent installation that uses an existing response file. The <i>response_file_path</i> specifies the absolute path and file name for the response file.</p> <p>For instructions, see Running a silent installation on page 17.</p>
-r <i>response_file_path</i>	<p>Creates a response file that is required for a silent installation. The <i>response_file_path</i> variable specifies the absolute path and file name for the response file. For more information, see Preparing a response file on page 16).</p> <p>Do not name your response file <code>bundle.properties</code> or <code>ids.properties</code>.</p>

Unsupported options

The installation user interface is based on the InstallAnywhere framework; however, the `onedb_install` command does not support the following options:

- `-add`
- `-remove`
- `-repair`
- `-uninstall`

To remove your installation, use the procedures that are recommended for your operating system.

Installation troubleshooting

Installation log files and return codes can provide helpful information about a completed installation or help you identify problems encountered during an installation attempt.

Tracing

Tracing is not enabled by default. You can enable tracing, set the level of tracing detail that is logged, and set the location of the debug files. To do so, include the `-DDEBUG_LEVEL` and `-DDEBUG_FILE` options with the [installation command on page 18](#).

Log files

When you install the database server, the installation application generates log files in the same directory as the installation media. You can change the default location of your log files and enable tracing during installation.

Installation and deployment log file

The installation and deployment log file, `iad_act.log`, is created by the installation application when you install products in the HCL OneDB™ software bundle.

- UNIX™, Linux™: `/tmp/iad_act.log`
- Windows™: `\tmp\iad_act.log`

Standard output and standard error log files

The `stdout` and `stderr` log files contain important information that you can use to debug installation failures.

- UNIX™, Linux™: `/tmp/bundle_install.stdout` and `/tmp/bundle_install.stderr`
- Windows™: `\tmp\bundle_install.stdout` and `\tmp\bundle_install.stderr`

Client products have separate log files.

InstallAnywhere log file

The InstallAnywhere log file contains information that is generated by the InstallAnywhere framework.

- UNIX™, Linux™: `$ONEDB_HOME/IBM_Informix_version_Install_date.log`
- Windows™: `%ONEDB_HOME%\IBM_Informix_version_Install_date.log`

If the installation application fails, the InstallAnywhere log files are placed in the following directories:

- UNIX™, Linux™: home directory
- Windows™: Desktop

InstallAnywhere return code (UNIX™, Linux™)

You can check the InstallAnywhere return code, also known as the *exit code*, after running the installation application to learn more about the installation status and possible error conditions. To retrieve the return code, run this command after the installation application finishes:

```
echo $?
```

For information about the InstallAnywhere return code, see the documentation at the Flexera Software web site: .

Creating a database server after installation

You create a database server by setting mandatory database server properties and then starting the database server.

To create a database server:

1. Configure the mandatory properties of the database server.
 - a. Set configuration parameters in the `onconfig` file.
 - b. Add connectivity information in the `sqlhosts` file and other connectivity files.
 - c. Set environment variables in your environment.



Tip: On Windows operating systems, you can use the Server Instance Manager to configure the mandatory properties of the database server instead of editing the `onconfig` and `sqlhosts` file and setting environment variables.

2. **Optional:** Configure the wire listener for MongoDB API and REST API support.
3. Initialize disk space for the database server when you start the database server for the first time. Disk space initialization creates the initial chunk of the root dbspace, which stores control and tracking information.



Attention: Do not initialize the disk space if you are upgrading from a previous version of the database server and you are using the same root dbspace. Initializing disk space deletes all existing user data and tracking information in the root dbspace.

Choose from:

- UNIX™, Linux™: Run the `oninit -i` command.
- Windows™: In the database server properties dialog box from the Services application, add `-i` in the **Start Parameters** field and click **Start**.

Results

The database server is started with disk space initialized.

Windows: A program group for the server is created from which you can open a command prompt for the database server, start the Server Instance Manager, uninstall the database server, or view documentation.

After you start the database server, you can set optional configuration parameters, environment variables, and connectivity settings.

Setting configuration parameters

You set configuration parameters that control database server behavior in the `onconfig` file. The `onconfig` file is a text file that you create based on a template file. The `onconfig` file contains many configuration parameters, but most have default values and can be set while the database server is running.

About this task

Use the template configuration file, `onconfig.std`, as a basis for your configuration file. The `onconfig.std` file contains most configuration parameters set to default values. The `onconfig.std` file is in the `$ONEDB_HOME/etc` directory (UNIX, Linux) or `%ONEDB_HOME%\etc` (Windows) directory.



Important: Do not modify or delete the `onconfig.std` file, which is a template and not a functional configuration.

To create an `onconfig` file and set the mandatory configuration parameters:

1. Copy the `onconfig.std` template file and save the file with a unique name, such as `onconfig.servername`, in the `$ONEDB_HOME/etc` directory (UNIX, Linux) or `%ONEDB_HOME%\etc` (Windows) directory.
2. Modify your `onconfig` file.

You can use several different methods to edit the `onconfig` file, including using a text editor.

At minimum, set the following configuration parameters:

- **DBSERVERNAME:** Set to the name of the database server.
- **ROOTPATH:** Set to the location of the root dbspace, which contains control and tracking information for the database server.
- **SERVERNUM:** Set to a unique number for the database server. If you do not have another database server on the same computer, you can leave the default value of 0.
- **LICENSE_SERVER:** Set to the License Server URL used to obtain the necessary licenses to run Informix

Results

When you set environment variables, set the **ONCONFIG** environment variable to the name of your customized configuration file.

If you omit a parameter value in your copy of the configuration file, the database server either uses default values from the `onconfig.std` template file or calculates values based on other parameter values.

Preparing connectivity files

Prepare the files that the database server uses to communicate with client applications and with other database servers.

About this task

The connectivity information allows a client application to connect to any HCL® OneDB® database server on the network. The connectivity data for a particular database server includes the database server name, the type of connection that a client can use to connect to it, the host name of the computer or node on which the database server runs, and the service name by which it is known.

You must prepare the connectivity information even if the client application and the database server are on the same computer or node. You are not required to specify all possible network connections in the `sqlhosts` file before you start the database server. However, after you add connectivity information to the `sqlhosts` file, you must restart the database server to make the new connection available.

To add connectivity information:

1. Edit the `sqlhosts` file to include the correct connectivity information with a text editor or equivalent tool.
 - The default name and location of this file is `$ONEDB_HOME/etc/sqlhosts` (UNIX, Linux) or `%ONEDB_HOME%\etc\sqlhosts` (Windows).
 - If you set up several database servers to use distributed queries, use either one `sqlhosts` file to which the `ONEDB_SQLHOSTS` environment variable points or separate `sqlhosts` files in each database server directory.
2. If your system uses Internet Protocol network connections, enter settings in the following files:

Choose from:

 - UNIX, Linux: The `/etc/hosts` and `/etc/services` files.
 - Windows: The `%windir%\system32\drivers\etc\services` file.

Setting environment variables

Set environment variables after you install the database server. If you created a database server during installation, all mandatory environment variables are set, however, you can set optional environment variables.

Before you begin

You must be logged in as an administrative user or with sufficient privileges to set environment variables.

About this task

UNIX, Linux: You can set environment variables at the command line, in an environment configuration file, or in a login file.

Windows: You can set environment variables in the system applet or at the command line.

To set the minimum required environment variables:

1. Set the `ONEDB_HOME` environment variable to the directory where you installed the database server.
2. UNIX, Linux: Set the `PATH` environment variable to include the `$ONEDB_HOME/bin` directory as follows.

- C shell:

```
setenv PATH ${ONEDB_HOME}/bin:${PATH}
```

- Bourne shell:

```
PATH=$ONEDB_HOME/bin:$PATH
export PATH
```

You must set the **ONEDB_HOME** variable and add `$ONEDB_HOME/bin` to the **PATH** environment variable for each user.

3. Set the **ONEDB_SERVER** environment parameter to the name of the database server.
4. Set the **ONCONFIG** environment variable to the name of a valid `onconfig` file that you created for the server.
5. If you want to use a locale or language other than the default locale of US English, set the following environment variables:
 - a. Set the **CLIENT_LOCALE** environment parameter to specify a nondefault locale.
 - b. Set the **DBLANG** environment parameter to specify the subdirectory of `ONEDB_HOME` that contains the customized language-specific message files that HCL® OneDB® products use.
 - c. Set the **DB_LOCALE** environment parameter.
 - d. Set the **SERVER_LOCALE** environment parameter.
 - e. Set the **GL_USEGLU** environment parameter if you plan to use UTF-8 character encoding. You must set the **GL_USEGLU** environment parameter before you create a database in which you plan to store UTF-8 character data.
6. Set the **ONEDB_SQLHOSTS** environment parameter to the `sqlhosts` file. The default location of this file is `$ONEDB_HOME/etc/sqlhosts` (UNIX, Linux) or `%ONEDB_HOME%\etc\sqlhosts` (Windows).
7. UNIX, Linux: Set the **ONEDB_TERM** environment parameter to specify whether the DB-Access utility uses the information in the `termcap` file or the `terminfo` directory.
On character-based systems, the `termcap` file and `terminfo` directory determine terminal-dependent keyboard and screen capabilities, such as the operation of function keys, color, and intensity attributes in screen displays, and the definition of window borders and graphic characters.
8. UNIX, Linux: If your applications are compiled with the HCL OneDB™ Client Software Development Kit (Client SDK), set the platform-specific library path environment variable to `$ONEDB_HOME/lib` and to the product library subdirectory. See the machine notes for your client products for the appropriate environment variable name.

Stopping and starting the database server

Various configuration tasks require you to stop and then start the database server. Shared memory is initialized every time that the database server starts.

About this task

The database server has various Database server operating modes .

- You can use the `onstat -` command to determine the database server mode.
- You can use the `onmode` utility to switch the database server from one mode to another.

Use the procedure for your operating system:

- [Stopping and starting the database server on page 25](#)
- [Stopping and starting the database server on page 25](#)

Stopping and starting the database server (UNIX™, Linux™)

You run the onmode utility to stop the database server, and the oninit utility to start the database server.

Before you begin

For a standard installation, log in as user **root** or user **informix**. For a non-root installation, log in as the owner of the database server.

1. To stop the database server, run the onmode -ku command.
2. To start the database server, run the oninit command.

You can include options to the oninit command. For example, you suppress verification prompts by running the oninit -y command.

Stopping and starting the database server (Windows™)

You stop and start the database server with the Services application.

Before you begin

Log in as a member of the Administrators group.

1. To stop the database server, in the server properties dialog box from the Services application, click **Stop**.
2. To start the database server, in the database server properties dialog box from the Services application, type optional oninit options in the **Start Parameters** box, and then click **Start**.

Alternatively, you can start the database server by running the starts command from a command prompt. You can also include options for the oninit command in the starts command.

Microsoft failover clustering support (Windows)

HCL OneDB™ database server supports Microsoft™ Failover Cluster (known as Microsoft™ Cluster Server, or MSCS, in earlier Windows™ releases). That built-in feature of the Windows™ operating system can automatically detect and respond to server or application failure, and can balance server workloads.



Important: During setup of the Microsoft™ failover cluster environment, stop the database server before you shut down the operating system to avoid data loss.

Refer to the Microsoft™ documentation for details about installation and configuration.

For a list of the Windows™ versions that HCL OneDB™ supports, go to [HCL OneDB™ Server System Requirements \(System requirements for HCL OneDB database server on page 5\)](#).

Multiresidency

Multiple residency refers to multiple database servers and their associated shared memory and disk structures that coexist on a single computer.

When you set up multiple independent database server environments on the same computer, you can do the following tasks:

- Separate production and development environments to protect the production system from the unpredictable nature of the development environment.
- Isolate sensitive applications or databases that are critically important, either to increase security or to accommodate more frequent backups than most databases require.

When you use multiple residency, each database server has its own configuration file. Thus, you can create a configuration file for each database server that meets its special requirements for backups, shared-memory use, and tuning priorities.

- Test distributed data transactions on a single computer. If you are developing an application for use on a network, you can use local loopback to simulate distributed data and testing on a single computer. Later, when a network is ready, you can use the application without changes to application source code.

When you plan for multiple residency on a computer, consider the following factors:

- Storage space

Each database server must have its own dedicated storage space. You cannot use the same disk space for more than one instance of a database server. When you prepare another database server, you must repeat some of the planning that you did to install the first database server. For example, you must consider these factors:

- UNIX, Linux: Whether the files are buffered or unbuffered, and whether the files share a disk partition with another application.
- The location of mirrored disks, if you plan to use mirroring.
- The location of the message log.
- The type of backups that you run.

- Shared memory

Each database server has dedicated shared memory. Your computer must have enough shared memory for all the database servers.

Running multiple database servers on the same computer is not as efficient as running one database server. You must balance the advantages of separate database servers with the extra performance cost.

Creating multiresident database servers (Windows)

You can set up multiple independent database server environments on the same computer.

Before you set up multiple residency, you must install one database server.



Important: You are not required to install more than one copy of the database server binary files. All instances of the same version of the database server on one computer can share the same executable files.

To set up multiple residency, use the Server Instance Manager. You can run the Server Instance Manager as a graphical program or on the command line.

To use the Server Instance Manager program, you must have administrative privileges on the database server as a member of the **Informix-Admin** group. However, local administrator privileges are sufficient, even if the database server was installed for domain use.

To create a new database server instance with the graphical Server Instance Manager program:

1. Start the Server Instance Manager from the database server program group on the **Start > All Programs** menu.
2. Choose an installation method, and click **Create New** to create a new instance of the database server.
3. Follow the prompts.

After you enter the required information, the Server Instance Manager installs services, records environment variables, updates the registry, and creates the `onconfig` and `sqlhost` files for the new database server instance.

Server instance manager command-line options (Windows)

The `instmgr.exe` utility is a command-line version of the Server Instance Manager that extends the ability to configure the HCL OneDB™ database server instance with command-line options.

You must have administrative privileges on the database server. However, local administrator privileges are sufficient, even if the database server was installed for domain use.

The following table describes the `instmgr.exe` utility options.

Table 5. The `instmgr.exe` utility options

Option	Meaning
<code>-alias DRDA_server_alias</code> <code>-drdaport DRDA_port drdasvc</code> <code>DRDA_service_name</code>	Command-line for stand-alone support of DRDA® in the DBMS.
<code>-apw password</code>	Specifies the user informix password for a new instance.
<code>-c -n servername -apw</code> <code>informix_password</code>	Creates an instance of the database server.
<code>-rename -apw</code> <code>informix_password -n</code> <code>new_name old_name</code>	Rename an existing server name.
<code>-s .ini_filename</code>	Initialize the specified instance in silent mode.

Table 5. The instmgr.exe utility options (continued)

Option	Meaning
-f <i>.ini_ filename</i>	Retrieve some information from a partial initialization file, and the remainder from the user.
-d -n <i>servername</i>	Delete the specified instance.  Warning: This option removes related dbspaces and deletes the data.
-dall	Delete all instances.  Warning: This option removes related dbspaces and deletes the data.
-uall -v <i>version</i>	Upgrade all instances to the specified version. (Upgrading a single instance in a multi-instance environment is not supported.)
-r -v <i>version</i> -n <i>servername</i>	Revert the specified instance to the specified version.
-rall -v <i>version</i>	Revert all instances to the specified version.
-b64 <i>Base64_password</i>	Specifies Base64 encoded password.
-l	Indicates cluster installation.
-system	Create database server instance that runs as local system user instead of user informix . The database server must be installed as local system user to use this option.

Creating multiresident database servers (UNIX™, Linux™)

You can set up multiple independent database server environments on the same computer.

Before you set up multiple residency, you must install one database server.



Important: You are not required to install more than one copy of the database server binary files. All instances of the same version of the database server on one computer can share binary files.

To create multiple residency of a database server:

1. Prepare a new configuration file and set the **ONCONFIG** environment variable to the new file name.
2. Set up connectivity for the new database server instance.
3. Initialize disk space for the new database server instance.
4. Prepare the backup environment for multiple residency.
5. Modify the operating system start to start the new database server instances automatically.
6. Check the **ONEDB_SERVER** environment variables for users.

Prepare a new configuration file

Each instance of the database server must have its own `onconfig` configuration file.

To set up an instance-specific `onconfig` file:

1. Make a copy of an `onconfig` file that has the basic characteristics that you want for your new database server.
2. Give the new file a name that you can easily associate with its function. For example, you might select the file name `onconfig.acct` to indicate the configuration file for a production system that contains accounting information.
3. Set the **ONCONFIG** environment variable to the file name of the new `onconfig` file. Specify only the file name, not the complete path.
4. In the new configuration file, set the following configuration parameters:

SERVENUM

Specifies an integer 0 - 255 that is associated with a database server configuration. Each instance of a database server on the same host computer must have a unique **SERVENUM** value.

DBSERVERNAME

Specifies the database server name of a database server. You can provide a useful name for the database server, such as **ifx12test** or *hostnameifx12test*.

MSGPATH

Specifies a unique path and name of the message file for a database server. If multiple database servers use the same **MSGPATH** parameter, you cannot identify the messages from separate database server instances. For example, if you name the database server **ifx12test**, you might specify `/usr/informix/ifx12test.log` as the message log for this instance of the database server.

ROOTPATH and ROOTOFFSET

Used together, specify the location of the root dbspace for a database server. The root dbspace location must be unique for every database server configuration.

If you put several root dbspaces in the same partition, you can use the same value for the **ROOTPATH** configuration parameter. However, in that case, you must set the **ROOTOFFSET** configuration parameter so that the combined values of the **ROOTSIZE** and **ROOTOFFSET** configuration parameters define a unique portion of the partition.

You are not required to change the value of the **ROOTNAME** configuration parameter. Even if both database servers have the name **rootdbs** for their root dbspace, the dbspaces are unique because each **ROOTPATH** configuration parameter specify a unique location.

You can also set the **MIRRORPATH** and **MIRROROFFSET** parameters. If the root dbspace is mirrored, the location of the root dbspace mirror must be unique for each database server.

Set up TCP/IP connectivity

If you use the TCP/IP communication protocol, you might be required to add an entry to the `services` file for the new database server instance.

The `sqlhosts` file must have an entry for each database server. If HCL® OneDB® products on other computers access this instance of the database server, the administrators on those computers must update their `sqlhosts` files.

If you plan to use Internet Protocol network connections with an instance of a database server, the system network administrator must update the `hosts` and `services` files.

Initialize disk space

Before you initialize disk space, check the setting of the **ONCONFIG** environment variable. If it is not set correctly, you might overwrite data from another database server. Initializing disk space deletes all existing user data and tracking information in the root dbspace. When you initialize disk space for a database server, the database server initializes the disk space that is specified in the current configuration file.



Important: As you create new blobspaces or dbspaces for a database server, assign each chunk to a unique location on the device. The database server prevents you from assigning more than one chunk to the same location within a single database server environment, but you must ensure that chunks that belong to different database servers do not overwrite each other.

To initialize disk space:

- UNIX™, Linux™: Run the `oninit -i` command.
- Windows™: In the database server properties dialog box from the Services application, add `-i` in the **Start Parameters** field and click **Start**.

Prepare the backup environment for multiple residency

Depending on your backup method, you must prepare the backup environment for multiple residency:

- If you use the ON-Bar utility, you can back up data from various database server instances to a single storage device if the storage manager allows it. The storage manager tracks what data is backed up. However, keep storage-space and logical-log backups on separate storage devices. A best practice is to keep storage-space and logical-log backups on separate storage devices.
- If you use the ontape utility, you must maintain separate storage space and logical log backups for each database server instance. If you can dedicate a tape drive to each database server, use the continuous logging option to back up your logical log files. Otherwise, you must plan your storage space and logical log backup schedules carefully so that use of a device for one database server instance does not cause the other database server instance to wait. You must reset the **ONCONFIG** environment variable every time that you switch backup operations from one database server instance to the other.

Modify operating system start for multiple server instances

You can ask your system administrator to modify the system start script so that each of your database server instances starts whenever the computer is rebooted; for example, after a power failure.

To start a second instance of a database server, change the **ONCONFIG** and **ONEDB_SERVER** environment variables to point to the configuration file for the second database server and then run the `oninit` command. Do not change the values of the **ONEDB_HOME** or **PATH** environment variables.

Similarly, you can ask the system administrator to modify the shutdown script so that all instances of a database server shutdown normally.

Reset the **ONEDB_SERVER** environment variable

If you want the new instance of a database to be the default database server, you must reset the **ONEDB_SERVER** environment variable.

Removing or modifying HCL OneDB™ installations

You can remove HCL OneDB™ products.

You can remove HCL OneDB™ products in the following ways:

- UNIX, Linux, Windows: You can remove the database server and all HCL OneDB™ client products that are installed in the same directory.
- You can remove the database server from a specific installation directory.

To modify an existing installation, remove the installation and then reinstall the product.

Removing or modifying installations (UNIX, Linux)

You can remove a database server and the HCL OneDB™ client products or remove only the database server.

You must be logged in as **root** or for non-root installations, as the owner of the database server.

If you have multiple installations of the product on the computer, the uninstallation application affects only the database server installation that is located in the `$ONEDB_HOME` path that you specify in the command. After HCL OneDB™ products are removed, you can manually delete the `$ONEDB_HOME` directory. It is not deleted automatically.

Removing an HCL OneDB™ software bundle installation

If you installed HCL OneDB™ client products, such as HCL OneDB™ Client Software Development Kit (Client SDK), OneDB® Connect, or the HCL OneDB™ JDBC Driver together in the same directory, you can quickly uninstall the database server and the client products at the same time.

To remove the database server and client product installations that are installed in the same folder, run one of the following commands:

- Console mode:

```
$ONEDB_HOME/uninstall/uninstall_server/uninstallserver -i console
```

- GUI mode:

```
$ONEDB_HOME/uninstall/uninstall_server/uninstallserver -i gui
```

Removing a database server

To remove a database server installation:

1. Run one of the following commands:

Console mode:

```
$ONEDB_HOME/uninstall/uninstall_ids/uninstallids -i console
```

GUI mode:

```
$ONEDB_HOME/uninstall/uninstall_ids/uninstallids -i gui
```

2. Follow the prompts in the uninstallation application.

Removing or modifying installations (Windows)

You can remove a database server and the HCL OneDB™ client products or remove only the database server. You can remove a database server in GUI mode or in silent mode.

You must have Windows™ administrator privileges to remove or modify the installation.

If you have multiple installations of the product on the computer, the uninstallation application affects only the database server installation that is in the `%ONEDB_HOME%` path that you specify in the command. After HCL OneDB™ products are removed, you can manually delete the `%ONEDB_HOME%` directory. It is not deleted automatically.

Removing an HCL OneDB™ software bundle installation

To remove the database server and all client products, start the uninstallation application by using one of the following methods:

- From the installation directory, run the `uninstallids.exe` executable file. For example, `%ONEDB_HOME%\uninstall\uninstall_ids\uninstallids.exe`.
- Select the program name in the Control Panel **Programs and Features** window and use the Uninstall or Change a Program utility to uninstall. For example, select **HCL OneDB™ Bundle *Version***, where *Version* is the database server version number, and then click **Uninstall**.

Removing a database server

To remove the database server:

1. Start the uninstallation application by using one of the following methods:
 - Select **Start > Programs > HCL OneDB™ Version > uninstallserver**, where *Version* is the database server version number.
 - Double-click the program name in the Control Panel **Programs and Features** window and run the Uninstall or Change a Program utility.
2. Follow the prompts in the uninstallation application.

Removing a database server in silent mode

You can create a response file by recording a GUI-based, interactive uninstallation of the database server from one location by using settings in the GUI application that you want to replicate. Then you run the silent uninstallation command, specifying the response file. A response file is not necessary if you do not have a customized installation.

To create the response file, you must have an installation that you can uninstall in a way that you plan to reuse for removal of other installations. It is not possible to do a silent uninstallation of HCL OneDB™ and any of the client products simultaneously.

To uninstall HCL OneDB™ in silent mode with a response file:

1. On a command line, start a GUI uninstallation of the database server with the `-x` option in the script:

```
uninstall\uninstall_server\uninstallserver -i gui -r response_file_name
```

Substitute *response_file_name* with full name and path of the response file. Append `.properties` at the end of the file name.

2. Copy the response file to the location where you want to uninstall the database server.
3. Run the silent uninstallation command, with the full path name of the response file:

```
uninstall\uninstall_server\uninstallserver -i silent -f response_file_name
```

Installing client products

These topics describe how to install HCL OneDB™ client products.

Installing HCL OneDB™ Client SDK

This topic describes how to install HCL OneDB™ Client SDK on UNIX™, Linux™, and Windows™ operating systems.

The *Quick Start Guide* provides a brief overview of installation. The installation instructions provide detailed instructions and alternative methods of installation.

Installing HCL OneDB™ Client Products

The *HCL OneDB™ Client Products Installation Guide* explains how to install HCL OneDB™ Client Software Development Kit (Client SDK).

This information is for database administrators or software engineers who install HCL® OneDB® client products. You must be familiar with the operating procedures of your computer and with your operating systems before installing these products.

These topics explain how to install Client SDK as individual products using stand-alone client installation applications. Alternatively, you can install Client SDK as part of the HCL OneDB™ software bundle installation media. The software bundle includes both the HCL OneDB™ database server and client products.

Preparing to install Client SDK

Before you install client products, you must prepare your system and plan for choices that you must make during the installation process.

About this task

Preparing for installation includes reviewing system requirements, deciding on the installation owner and the installation directory, and reviewing installation options. It also includes preinstallation tasks such as obtaining the software and reviewing the machine notes.

To prepare for installation:

1. Obtain the software.
For more information about HCL OneDB™ see, <https://www.hcltechsw.com/products/onedb>.
2. Review the client components that you can install.
You can install all of them or some of them.
3. Ensure that your computer meets the [system requirements on page 37](#).
4. Choose a user account with which to install the product. The user account becomes the installation owner.
 - UNIX, Linux: Decide between a non-root owner and an owner with root or administrative privileges.
 - If you plan to install as an administrative user, decide on a password for user **informix**.
5. Choose an installation directory.
If client products are already installed in the directory, uninstall the existing client products.
6. Choose an installation method.
7. If you are installing client products on the same computer as the database server, install or upgrade the database server before you install client products.

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

What to do next

After you install client products, you can configure the environment and connect to the database server.

HCL OneDB™ Client Software Development Kit components

You use the Client SDK to develop and run client applications. Client SDK includes APIs, programs, and database utilities.

Applications that run on client computers require HCL® OneDB® Connect to access database servers. OneDB® Connect is a runtime connectivity product that is composed of runtime libraries that are included in Client SDK.

The following table describes the APIs and programs of Client SDK.

Table 6. Client SDK APIs and programs

Component	Description
HCL® OneDB® .NET Core Provider (Windows™ only)	<p>The HCL® OneDB® .NET Core Provider is a .NET assembly that lets .NET applications access and manipulate data in HCL® OneDB® databases. It does this by implementing several interfaces in the Microsoft™ .NET Framework that are used to access data from a database. Using the HCL® OneDB® .NET Core Provider is more efficient than accessing the HCL OneDB™ database through either of these methods:</p> <ul style="list-style-type: none"> • Using the Microsoft™ .NET Framework Data Provider for ODBC along with the HCL OneDB™ ODBC Driver • Using the Microsoft™ .NET Framework Data Provider for OLE DB along with the HCL OneDB™ ODBC Driver
HCL OneDB .NET Core Provider (Windows and Linux)	<p>The HCL® OneDB® .NET Core Provider is a .NET assembly that lets .NET applications access and manipulate data in IBM Informix databases. It does this by implementing several interfaces in .NET Core SDK/Runtime v3.1 which is shipped for Windows x64 and Linux x86_64 platforms. Name of all the Interfaces/Methods in HCL® OneDB® .NET Core Provider are same as .NET Framework v4.x Provider. HCL OneDB .NET Core adds few More Functionality in addition to .NET Framework v4.x Provider.</p>
with XA support	<p>An SQL embedded-language product that is used to create custom C applications.</p> <p>For more information, see the ESQL/C Guide on page .</p> <p>A C++ interface to develop object-oriented client applications for use with all HCL OneDB™ database servers and client-side value objects for HCL OneDB™</p> <p>For more information, see the OneDB® Object Interface for C++ Guide on page .</p>
HCL OneDB™ ODBC Driver with MTS support	<p>The HCL OneDB™ implementation of the Open Database Connectivity (ODBC) 3.0 Level 1+ standard that supports Microsoft™ Transaction Server (MTS). This driver contains extensibility support for HCL OneDB™.</p> <p>For more information, see the HCL OneDB™ ODBC Driver Guide on page .</p>
HCL® OneDB® OLE DB Provider (Windows™ only)	<p>A client-side, native OLE DB provider that implements full functionality for base-level providers and contains extensibility support for HCL OneDB™.</p> <p>For more information, see the HCL OneDB™ OLE DB Provider Guide on page .</p>

The following table describes the HCL OneDB™ common database utilities in the Client SDK.

Table 7. HCL OneDB™ common database utilities

Component	Description
Documentation Viewer	Use the viewer to see the release and machine notes in text format.
The finderr utility on UNIX™ systems and the HCL OneDB™ Error Messages utility on Windows™ systems	Use these utilities to obtain information about error messages, including corrective actions, that are specific to HCL OneDB™.
The ILogin utility (Windows™ only)	Use the ILogin Demo utility to quickly test the connection to HCL® OneDB®. Click File > Run then enter the HCL OneDB™ connection parameters. You must create the target database before you connect to it.
DB-Access	<p>The DB-Access utility is included in the HCL OneDB™ Client SDK when the corresponding HCL OneDB™ server version is available for the same operating system. When the DB-Access utility is included in the Client SDK, it is the same utility that is delivered with the HCL OneDB™ server. The utility provides a stand-alone client command-line tool that can directly access the HCL OneDB™ database instance. This dbaccess client supports the same user interface and functionality as the dbaccess server utility.</p> <p>For the setup requirements, see Requirements for the HCL OneDB™ Client Software Development Kit DB-Access utility on page .</p>
The ConnectTest utility (Windows™ only)	<p>Use the ConnectTest utility to test the connection to the HCL OneDB™ server. You can start the utility from the shortcut in the Start menu or from the <code>\$INFORMIX/bin</code> directory. The utility populates the server details from the registry that is registered by the Setnet32 utility. You can manually edit the utility to connect to other servers not listed in the registry, which does not modify the registry entry. The utility shows the resulting data of the executed SQL query.</p>
Setnet32 (Windows™ only)	Use the Setnet32 utility for configuring client products. Each package might include online help, example programs, and support programs.

System requirements for HCL OneDB™ client products

This section provide detailed information about the system requirements for HCL OneDB™ client products.

Hardware requirements

Before you install client products, make sure that your computer meets the hardware requirements.

Disk space and memory requirements for installation

The following disk space and memory requirements for installing the client products are approximate. The requirements for some operating systems might be lower.

Disk space

Client SDK: approximately 200 MB

Some installation choices require additional disk space. The installation application informs you of the total disk space required by your setup before you copy the binary files to your computer.

RAM

UNIX, Linux: 512 MB

Windows: 256 MB

Temporary disk space

You can change the location of the temporary directory by setting the **IATEMPDIR** environment variable before you run the installation application.

UNIX™, Linux: 1 GB free space is required in your temporary directory (`/tmp` by default). If the temporary directory does not have enough space, the installation application attempts to write the information into the home directory of the user account.

Windows™: 1 GB free space is required in your temporary directory (`C:\tmp` by default).

Restrictions

Linux: Client SDK does not support RPM installations.

Software Requirements

Before you install the Client SDK, make sure that your computer meets these software requirements.

Operating system requirements

Install the required operating system patches and library files that are listed in the machine notes for your operating system. The machine notes are available in the doc directory of the media files before installation.

UNIX™, Linux™: The directory on which you plan to install the database server must be local or an NFS-mounted file system with regular operating-system files.

Windows™: The drive on which you plan to install the database server must be formatted with NTFS and must have 8.3 file name creation enabled. Turn off User Account Control (UAC) security before you install the database server.

HCL OneDB™ Client SDK and Server 1.0 is certified to run on 64-bit versions of these operating systems:

Table 8.

Operating System	Version
Linux x86-64	RHEL 7.4 / CentOS 7.4

Table 8. (continued)

Operating System	Version
	<ul style="list-style-type: none"> • Base kernel version 3.10.0 • Glibc version 2.17 • Compiler 4.8.5 <p>RHEL 8</p> <ul style="list-style-type: none"> • Base kernel version 4.18.0 • Glibc version 2.28 • Compiler 8.2.1 <p>SuSE SLES 12</p> <ul style="list-style-type: none"> • Minimum service pack 3 • Base kernel version 4.4.73 • Glibc version 2.22 • Compiler 4.8.5 <p>SuSE SLES 15</p> <ul style="list-style-type: none"> • Base kernel version 4.12.14 • Glibc version 2.26 • Compiler 7.3.1 <p>Ubuntu 16.04 LTS</p> <ul style="list-style-type: none"> • Base kernel version 4.4.0 • Glibc version 2.23 • Compiler 5.4.0 <p>Ubuntu 18.04 LTS</p> <ul style="list-style-type: none"> • Base kernel version 4.15 • Glibc version 2.27 • Compiler 7.5 <p>Ubuntu 20.04 LTS</p> <ul style="list-style-type: none"> • Base kernel version 5.4 • Glibc version 2.31 • Compiler 9.3
AIX	7.2

Table 8. (continued)

Operating System	Version
Windows	Windows 2016 Windows 2019 Windows 10

Application software required to run/install OneDB ClientSDK

Table 9.

Operating System	Version
Linux	OpenSSL any of the supported versions 1.0.2, 1.1.0 or 1.1.1.
AIX	OpenSSL 1.0.2 is required
Java8	JRE is required to install OneDB. Some components of OneDB also require Java8 Windows install requires a 1.8.0 JRE Linux/AIX install requires a 1.8.0 or later JRE

Installation methods

The primary way to install Client SDK or OneDB® Connect is with an interactive installation that prompts you to configure installation properties. After you run an interactive installation, you can run non-interactive installations that are based on the original installation.

Interactive installation

An interactive client products installation allows you to exclude unneeded products and to prepare for a silent installation by creating a response file. After you start the installation program, choose a custom installation to exclude products, or a typical installation to install all components.

Non-interactive installation methods

Non-interactive installation methods, such as a silent installation or a script-based installation, are based on the results of interactive installation or on an existing client products installation.

Silent installation (UNIX, Linux, Windows)

Use to install Client SDK or OneDB® Connect, with the product-specific command.

By using a response file, you can replicate a specific installation configuration on multiple computers with minimal or no user interaction. Silent installation can save much time for some user scenarios.

Extraction with command-line script (UNIX, Linux)

Use to install Client SDK or OneDB® Connect individually.

The extraction with command-line installation alternative can be helpful in one of the following scenarios:

- You want to install the product to redistribute it to multiple computers quickly and with minimal disk space usage
- You want to avoid using Java™ Runtime Environment (JRE) during redistribution of the product

Installation owner

The user who installs the client products is the owner of the installation.

The standard way to install client products is as the superuser **informix** with administrative privileges.

UNIX, Linux

You can install client products as the superuser **informix** or as a non-root user.

User **informix** is required for root-based installations because it has the unique user identifier (UID) to manage client products, as well as the database server. The installation program automatically creates the user **informix** and the group **informix**. If you previously installed a root-based HCL OneDB™ product on the computer, user and group **informix** already exist.

You can install the client products as a non-root user. Then the user account that performs the non-root installation is the owner of the installation. However, you cannot perform a non-root installation of a client product in an **\$ONEDB_HOME** directory that has a standard, root-based installation of the HCL OneDB™ server.

If you are installing client products on a platform that does not support Java, you must manually create the user **informix** and the group **informix**. For instructions, see [Creating the group informix and user informix \(UNIX, Linux\) on page 9](#).

Windows

You must have Administrator privileges to install client products. If you log on to Windows™ without specifying a domain, only the local system is checked. The user **informix** is created during the installation as a member of the Administrator group.

Installation directory

You can use the default installation directory for the database server when you run the installation application, or you can create a directory before you install and then select that directory during installation.

The installation directory is known as the **ONEDB_HOME** directory because the environment variable **ONEDB_HOME** is set to the installation directory.

The directory where you install HCL OneDB™ products must fulfill these requirements:

- The directory can contain a database server installation.
- The directory must not contain HCL OneDB™ client products installations. Uninstall existing client products before you install new client products in the same directory.
- The full path to the installation directory must not exceed 200 characters, including path separators, and must not contain spaces. UNIX™, Linux: The directory for a non-root installation must not exceed 60 characters.
- UNIX™, Linux™: The directory on which you plan to install the database server must be local or an NFS-mounted file system with regular operating-system files.
- Windows™: The drive on which you plan to install the database server must be formatted with NTFS and must have 8.3 file name creation enabled. For Windows™ 7 and subsequent Windows™ versions, turn off User Account Control (UAC) security before you install client products.



Tip: UNIX™, Linux: Do not use *back* or *quit* as a string in the **ONEDB_HOME** path or in any HCL OneDB™ product installation configuration settings, including passwords.

Installing client products

You can install client products as an interactive installation or an unattended installation. You can customize the installation process with the installation command.

You have the following installation method options, depending on your operating system:

- UNIX, Linux: interactive console (default), interactive GUI, silent, scripted
- Windows: interactive GUI, silent

Although the most common way to run the installation program on Windows or Mac OS X operating systems is by double-clicking an executable, you can run the installation command from the command line on all operating systems. When you run the installation command from the command line, you can include options to customize the installation process in the following ways:

- Specify the mode for the installation program: console, GUI, or silent, depending on the operating system.
- Create a response file for use in an unattended installation.
- Specify the directory for installation log files.
- Specify the level and directory for debugging information.
- Specify silent installation options.

Running an interactive installation

You can install client products by running an interactive installation program. You can choose which products to install.

Before you begin

Before you start the installation program, perform the following steps:

- [Prepare your system for installation. on page 35](#)
- Log in as the appropriate user to be the owner of the client products.

About this task

You can include options to the installation command if you run the command from the command line. For example, on UNIX and Linux, you can include the `-i gui` option to run the installation program in GUI mode instead of the default console mode.

To install client products:

1. Start the installation application from the media directory.
 - UNIX, Linux:
 - a. If necessary, extract the product files. For example, run the tar command:


```
tar xvf filename
```

The *filename* is the name of the product tar file.
 - b. Run the following command to install Client SDK: `./installclientsdk`.
 - Windows:
 - a. If the entire media is still in .zip archive format, extract the media.
 - b. Exit all other applications.
 - c. Double-click the `installclientsdk.exe` file to install Client SDK.
2. Follow the instructions in the installation application. Choose a custom installation to exclude products, or a typical installation to install all components.
3. Complete the installation and exit the installation application.

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

You can now configure the client environment and connect to the database server.

What to do next

Windows: If you are upgrading to new release, restart your computer after the installation completes if both of the following conditions are true:

- The new installation location is different from the previous installation location
- You plan to use a distributed transaction application that uses the COM+ technology

Running a silent installation

To perform a silent installation, which is sometimes referred to as an *unattended installation*, you must create a response file that contains information about how you want the product installed. You invoke this response file in a command-line option to perform the silent installation.

Before you begin

Before you start the installation program, perform the following steps:

- [Prepare your system for installation. on page 35](#)
- Log in as the appropriate user to be the owner of the client products.

About this task

On each target computer, you can customize the silent installation command in the following ways:

- Accept the license. Include the `-DLICENSE_ACCEPTED=TRUE` property.
- Bypass operating system prerequisite checks. Use with caution. Include the `-DPRQCHECK=FALSE` property.
- Set the installation path. Include the `-DUSER_INSTALL_DIR=path` property.
- Overwrite existing product installations. Use with caution. Include the `-DOVERWRITE_PRODUCT=TRUE` property.

To deploy Client SDK in multiple directories:

1. Create a response file by doing one of the following actions:

Choose from:

- On a command line, run the appropriate installation command with the `-r` option to start an interactive installation that generates a response file.
 - Edit the `csdk.properties` file that is on the installation media, rename the file, and configure the settings for your environment and agreement to the license terms.
2. Copy the response file and the installation media to the computer where you want to install the client products.
 3. Run the silent installation command for the client products that you want to install, indicating the absolute path to the response file after the `-f` option:

UNIX, Linux:

```
./installclientsdk -i silent -f path_name
```

Windows:

```
./installclientsdk.exe -i silent -f path_name
```

Results

Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

You can now configure the client environment and connect to the database server.

What to do next

Windows: If you are upgrading to new release, restart your computer after the installation completes if both of the following conditions are true:

- The new installation location is different from the previous installation location
- You plan to use a distributed transaction application that uses the COM+ technology

Redistributing client products with scripts (UNIX, Linux)

You can extract client product files from the installation media for script-based redistribution to other computers.

Before you begin

Complete the following prerequisite tasks:

- [Prepare your systems for installation on page 35.](#)
- Log in as **root** user to create and copy a root installation that runs with user and group **informix** accounts.

About this task

Use this method if you want to redistribute the client product files without adding a Java™ runtime environment or an uninstallation application to multiple host systems. You extract the product files by starting the installation application with the `-DLEGACY=TRUE` command on one computer. After you run the installation application, copy the files in the installation directory of the first computer to redistribute the product on other computers.

To install client products with a script:

1. Run an installation command:

```
./installclientsdk -DLEGACY=TRUE
```

2. Follow the instructions in the installation application.
3. Copy the contents of the installation directory and place them into the directory of another computer where you want to redistribute the products.
The contents include the RUNasroot scripts that you use for redistribution.
4. Run the script for the product that you want to deploy on the other computer.

```
RUNasroot.installclientsdk
```

Results

Completes the redistribution of the Client SDK files. Log files are created during the installation process. You can use the log files to troubleshoot installation errors.

You can now configure the client environment and connect to the database server.

Client products installation command syntax

You can include options to the installation command when you run the command from the command line..

Log in as the appropriate user to be the owner of the database server to run this command.

Syntax

```
(1)
>>+-installclientsdk-----+-- -i----->
|                               (2) |
+-installclientsdk.exe-----+
|                               (1)  |
+-installconnect-----+
|                               (2)  |
'-installconnect.exe-----'

(1)
```


Table 10. Installation options (continued)

Option	Meaning
-DLICENSE_ACCEPTED=TRUE	Accepts the product license terms during a silent installation instead of indicating acceptance in the response file.
-DOVERWRITE_PRODUCT=TRUE	Specifies to overwrite an existing product installation during a silent installation. Otherwise, the installation application exits if it finds an existing product installation.
-DPRQCHECK=FALSE	<p>Specifies to bypass the operating system prerequisites check during a silent installation.</p> <p> Important: Use with caution. Disabling the prerequisites check might result in an installation that does not function properly.</p>
-DUSER_INSTALL_DIR= <i>install_path</i>	Specifies the installation path during a silent installation instead of specifying the path in a response file. Replace <i>install_path</i> with the absolute path to the installation directory.
-help	Displays list of supported options and their functions.
-i gui	Starts the installation program in GUI mode.
-i console	UNIX™, Linux™: Starts the installation program in console mode.
-i silent -f <i>response_file_path</i>	<p>Starts a silent installation that uses a response file. The <i>response_file_path</i> specifies the absolute path and file name for the response file.</p> <p>For instructions, see Running a silent installation on page 43.</p>
-r <i>response_file_path</i>	<p>Creates a response file from the installation. The <i>response_file_path</i> specifies the absolute path and file name for the response file. The response file contains an installation configuration that you want to deploy in more than one location.</p> <p>For more information, see Running a silent installation on page 43.</p> <p>Do not name your response file <code>bundle.properties</code> or <code>ids.properties</code>.</p>

Unsupported options

The installation user interface is based on the InstallAnywhere framework; however, the `installclientsdk` and `installconnect` commands do not support the following options:

- -add
- -remove
- -repair
- -uninstall

To uninstall or modify client products, use the procedures that are recommended for the client products and operating systems.

Post-installation tasks for client products

After you install client products, you can configure your environment and create connections to the database server.

The method for creating connections depends on the client API or program. In some cases, you edit the `sqlhosts` file and other connectivity files. For instructions, see the documentation for the specific API.

Environment variables for clients

Client SDK, supports the following environment variables.

For JDBC-specific environment variables, see [Informix environment variables with the IBM Informix JDBC Driver on page](#).

The following environment variables affect all client products, unless otherwise stated.

Table 11. Environment variables for Client SDK

Environment variable	Purpose	Publication reference and possible values
CC8BITLEVEL on page	Specifies how the C-language compiler processes non-ASCII (8-bit and multibyte) characters.	Default is 2.
CLIENT_LOCALE on page	Specifies the locale that the client application uses to perform read/write operations that involve the keyboard, display, file, or printer on the client computer.	Any valid locale specifier. No default.
DB_LOCALE on page	Specifies the locale of all the databases that a client application accesses in a single connection (the database locale).	
DBANSIWARN on page	When turned on (y), causes the preprocessor to check for HCL OneDB™ extensions to ANSI-standard SQL syntax at compile time and causes an application to check for HCL OneDB™ extensions at run time.	Default is n.

Table 11. Environment variables for Client SDK (continued)

Environment variable	Purpose	Publication reference and possible values
DBCENTURY on page	Specifies how to expand values of two-digit-year DATE and DATETIME values. <code>P</code> = past century, <code>R</code> = present century, <code>C</code> = closest century.	Values: <code>P</code> , <code>R</code> , or <code>C</code> . Default is <code>R</code> .
DBFLTMASK on page	Specifies the number of decimal digits to use when storing a DECIMAL, SMALLFLOAT, or FLOAT data type in a character buffer DB-Access only.	Values: 0–16. Default is 16.
DBLANG on page	Specifies the subdirectory of the HCL OneDB™ installation directory that contains the product-specific message (<code>.iem</code>) files.	Default is <code>\msg</code> .
DBMONEY on page	Specifies the end-user format for MONEY values so that they can conform with various international and local monetary conventions.	Default is <code>\$,.,.</code>
DBPATH on page	Identifies the database servers that contain databases that the application accesses.	
DBTEMP on page	Specifies the directory on the client computer that the client processor uses to store temporary files (also called <i>swap</i> files).	Default is <code>\tmp</code> .
DELIMIDENT on page	Indicates whether (y) or (n) to interpret strings in double quotation marks as delimited identifiers.	Default is <code>n</code> .
ESQLMF on page	Indicates whether the ESQL/C processor automatically invokes the ESQL/C multibyte filter (<code>value=1</code> and <code>CC8BITLEVEL<3</code> , or not <code>value=0</code>).	
FET_BUF_SIZE on page	Overrides the default size of the fetch buffer.	Any valid buffer size, in bytes. Default value depends on row size.
GL_DATE on page	Specifies a customized end-user format for DATE values.	
GL_DATETIME on page	Specifies a customized end-user format for DATETIME values.	
IFX_DISABLE_IPV6 on page	Disables IPv6 support for a single database instance or for a single client application.	Values: <code>yes</code> or <code>no</code> . Default is <code>no</code> .

Table 11. Environment variables for Client SDK (continued)

Environment variable	Purpose	Publication reference and possible values
IFX_FLAT_UCSQ on page	Overrides the global setting and directs the optimizer to use subquery flattening for all sessions.	Default is <code>0</code> .
CONNECT_RETRIES on page	Specifies the maximum number of additional connection attempts made to a database server in the time limit that CONTIME specifies.	Default is <code>1</code> .
CONNECT_TIMEOUT on page	Specifies the number of seconds an SQL CONNECT statement continues to try to establish a connection before it generates an error.	Default is <code>60 (seconds)</code> .
ONEDB_HOME on page	Identifies the location of the client programs, library files, message files, header files, and other HCL® OneDB® software components. ONEDB_HOME must always be set.	Default is <code>C:\INFORMIX</code> .
ONEDB_SERVER on page	Identifies the default database server.	
ONEDB_SQLHOSTS on page	Specifies the name of the computer on which the central registry is located.	
NODEFDAC on page	In databases not created WITH LOG MODE ANSI, setting this to <code>yes</code> prevents the PUBLIC group from receiving table privileges or routine privileges by default when tables or when owner-privileged UDRs are created.	Default is for PUBLIC to hold all table and routine privileges.
OPTMSG on page	Enables optimized message transfers (message chaining) for all SQL statements in an application.	Default is the value that LANG specifies.
OPTOFC on page	Enables optimize-OPEN-FETCH-CLOSE functionality in an ESQL/C application that uses DECLARE and OPEN statements to execute a cursor.	Default is the value that LANG specifies.
TMP	Can be used to change the location temporary space directory used during installation. This is useful if there is not enough space in the default <code>/tmp</code> directory.	Any valid path.

Configuring client products on Windows™ systems

You can use the Setnet32 utility to configure Client SDK products.

The Setnet32 utility sets or modifies environment variables and network parameters that HCL OneDB™ products use at run time. You can access the Setnet32 utility from the client products program group.

You can set the following types of information with the Setnet32 utility:

- Environment variables. The environment variables and network parameters are stored in the Windows™ system registry and are valid for every HCL OneDB™ client product that you install.

To set component-specific environment variables, see individual product documentation.

- Database server network information. A client application connects to a database server that is running on a computer that can be reached through the network. To establish the connection, use Setnet32 to specify the location of the database server on the network and the network communications protocol to use. You must obtain this information from the administrator of the database server you want to use.



Requirement: If you enter a service name, it must be defined on the client computer in the `system32\drivers\etc\services` file in the Windows™ installation directory. The service definition must match the definition on the database server host computer.

- Host computer and login information. A client application can make connections only to a host computer that it can access through the network. You describe a connection to a host computer by specifying host parameters.

You can use a set of environment variables with the function call `ifx_putenv` to override the information that was entered in the **Server Information** and **Host Information** pages of the Setnet32 utility.

To use these environment variables, set the **ONEDB_SERVER** environment variable to a valid database server name. Overrides affect only that database server.

The following table lists the WIN32 override environment variables.

Table 12. WIN32 Override Environment Variables

Environment Variable	Overrides the Value For
WIN32HOST	The HOST network parameter
WIN32USER	The USER network parameter
WIN32PASS	The PASSWORD network parameter
WIN32ASKPASSATCONNECT	The PASSWORD OPTION network parameter
WIN32SERVICE	The SERVICE network parameter
WIN32PROTOCOL	The PROTOCOL network parameter

Connectivity protocols

The connectivity protocols that you can use to connect client products to the database server depend on the operating system.

The following table uses these abbreviations for protocol names:

- BSTP - Berkeley sockets using TCP/IP
- IPSP - IPC using stream pipe
- IPSM - IPC using shared memory
- IPNM - IPC using a named pipe
- TLTP - TLI using TCP/IP
- TLIX - TLI using IPX/SPX

Table 13. Connectivity protocols for client products

Platform/OS	BSTP	IPSP	IPSM	IPNM	TLTP
AIX	onsoctcp drsoctcp	onipcstr	onipcshm		
HP Itanium™ 64	onsoctcp drsoctcp	onipcstr	onipcshm		
HPUX	onsoctcp drsoctcp	onipcstr	onipcshm		
Linux™ 32-bit	onsoctcp drsoctcp	onipcstr	onipcshm		
Linux™ AMD 64	onsoctcp drsoctcp	onipcstr	onipcshm		
Linux™ Power	onsoctcp drsoctcp	onipcstr	onipcshm		
Linux Power Little Endian	onsoctcp drsoctcp	onipcstr	onipcshm		
Linux zSeries	onsoctcp drsoctcp	onipcstr	onipcshm		

Table 13. Connectivity protocols for client products (continued)

Platform/OS	BSTP	IPSP	IPSM	IPNM	TLTP
Linux ARM	onsoctcp drsotcp	onipcstr	onipcshm		
Solaris	onsoctcp drsotcp	onipcstr	onipcshm		ontlitcp
Windows	onsoctcp drsotcp			onipcnmp	

**Notes:**

- If you are using 64-bit servers: Remote 32-bit applications/tools (example: ESQL/C) can only connect to HCL OneDB™ using the TCP/IP protocol. Local 32-bit applications/tools can connect to the 64-bit server using either TCP/IP protocols or IPC stream pipe protocols. Local 32-bit applications/tools cannot connect to the 64-bit server using IPC shared memory protocol.
- If you are using a NUMA system and have several IP addresses configured on the same physical interface: The HCL OneDB™ TCP listener thread binds to a particular IP address/port. It listens for incoming calls addressed to only one IP address. If there are multiple IP addresses configured on the system, a TLI client is not able to connect to the server. (If DNS is configured to resolve the host name on a rotary of the four IP addresses, you do not know which of those addresses has been selected by the HCL OneDB™ TCP listener). The solution is that in the `$ONEDB_HOME/etc/sqlhosts` file, use `"*hostname"` instead of `"hostname"`. It will make the HCL OneDB™ TCP listener bind to `INADDR_ANY` instead of a particular address.
- For additional information, see the HCL OneDB™ machine notes.

Removing client products

How you uninstall client products depends on the operating system and the installation method.

You have the following product removal methods:

- UNIX, Linux: interactive console, interactive GUI, silent, manual (only for products installed by a script)
- Windows: interactive GUI

Removing client products

You can uninstall HCL OneDB™ client products with an uninstallation application in GUI, console, or silent mode on UNIX™, Linux™. Which mode you select depends on your objectives and system environment.

Before you begin

You must be logged in as a **root** (UNIX, Linux) or an administrative user (Windows) to perform this task for root-based installations.

For non-root installations, either the installation owner or root user can remove the product. If you want to remove the product as a root user, verify that the installation is not in use or in production before uninstalling.

About this task



Important:

- Do not remove any HCL OneDB™ products by manually deleting files.
- If Client SDK is installed in the same directory as the HCL OneDB™ database server:
 - You must uninstall HCL OneDB™ before you uninstall Client SDK.
 - Do not remove because both products have dependencies on this feature.

To uninstall client products:

1. From the **ONEDB_HOME** directory, run one of the following commands from the command line:

- UNIX, Linux:

Client SDK: `uninstall/uninstall_csdk/uninstallcsdk`

HCL OneDB™: `uninstall/uninstall_connect/uninstallconnect`

Include the `-i swing` or the `-i gui` option to uninstall in GUI mode.

- Windows:

Client SDK: `uninstall\uninstall_csdk\uninstallcsdk.exe`

HCL OneDB™: `uninstall\uninstall_connect\uninstallconnect.exe`

Alternatively, you can choose the uninstallation option from the client products program group. You cannot use the **Add/Remove Programs** utility in the Control Panel to remove HCL OneDB™ client products.

2. Follow the prompts to complete the uninstallation.

Results

Log files are created during the uninstallation process. You can use the log files to troubleshoot uninstallation errors.

Running a silent uninstallation (UNIX, Linux)

You can run a silent uninstallation of Client SDK and HCL OneDB™ on UNIX and Linux operating systems.

Before you begin

You must be logged in as a **root** user.

About this task

Silent uninstallation of Client SDK and OneDB® Connect does not require a response file. A response file is required only if you want to customize the uninstallation. The only way to create the response file for silent uninstallation of a client product is to record a GUI or console uninstallation of the product in the first directory where you want to remove it. You cannot create the response file by modifying the template `bundle.properties` file.

To uninstall client products in silent mode:

1. **Optional:** If you want to create a response file, select a product installation that you want to uninstall with customized settings that are reusable for uninstallation of the same product on other computers.
2. **Optional:** Start a Client SDK or HCL OneDB™ Connect product uninstallation application in GUI or console mode and specify the `-i -r path_name` option on the command line, replacing `path_name` with a full path to the response file that you want to record.

Example

For example, if you want to uninstall HCL OneDB™ Connect in console mode and record the uninstallation in a response file at `C:/informix/uninstall.ini`, you would run:

```
$ONEDB_HOME/uninstall/uninstall_connect/uninstallconnect -i -r uninstall.ini
```

If you want to uninstall Client SDK, use the `uninstall/uninstall_csdk/uninstallclientsdk` command. The `-i -r path_name` option can be used together with the `-gui` option in the same command line if you prefer to perform the uninstallation in GUI mode.

3. **Optional:** Copy the response file to the computer where you want to uninstall the same client program.
4. Run the uninstallation command for the product with the `-i silent` option. If you are using a response file, indicate the relative or absolute path to the response file after the `-f` option.



Attention: Do not pass the `-f path_name` argument shown in the following example if you are not using a response file.

Choose from:

- Client SDK: `uninstall/uninstall_csdk/uninstallclientsdk -i silent -f path_name`
- OneDB® Connect: `uninstall/uninstall_connect/uninstallconnect -i silent -f path_name`

Removing a scripted installation (UNIX™, Linux™)

If you ran the command-line script to install Client SDK, you must remove the client product files manually.

About this task

If you installed the database server with scripts on the same computer, removing client products also removes the database server. To remove the HCL OneDB™ files manually, run the following command from the `ONEDB_HOME` directory as the `root` user:

```
rm -rf productfiles
```

Client installation troubleshooting

Installation log files and return codes can provide helpful information about a completed installation or help you identify problems encountered during an installation attempt.

Log files

When you install the database server, the installation application generates log files in the same directory as the installation media. You can change the default location of your log files and enable debugging mode during installation.

Install and Deployment log file

The Install and Deployment log file `iad_act` is created for any installation by using the HCL OneDB™ software bundle.

- UNIX™, Linux™: `/tmp/iad_act`
- Windows™: `\tmp\iad_act`

Standard-Out and Standard-Error log files

The Standard-Out and Standard-Error log files contain important debug information for use during installation failures.

- UNIX™, Linux™: `/tmp/bundle_install.stdout` and `/tmp/bundle_install.stderr`
- Windows™: `\tmp\bundle_install.stdout` and `\tmp\bundle_install.stderr`

InstallAnywhere log file

The InstallAnywhere log file contains information about the installation by using the InstallAnywhere framework.

- UNIX™, Linux™:
 - Client SDK installation: `$ONEDB_HOME/IBM_Informix_Client-SDK_version_Install_date.log`
 - OneDB® Connect installation: `$ONEDB_HOME/IBM_Informix_Connect_version_Install_date.log`
 - Client SDK installation that includes Global Language Support (GLS): `$ONEDB_HOME/IBM_Informix_GLS_version_Install_date.log`
- Windows™:
 - Client SDK installation: `%ONEDB_HOME%\IBM_Informix_Client-SDK_version_Install_date.log`
 - OneDB® Connect installation: `%ONEDB_HOME%\IBM_Informix_Connect_version_Install_date.log`
 - Client SDK installation that includes Global Language Support (GLS): `%ONEDB_HOME%\IBM_Informix_GLS_version_Install_date.log`

If the installation application fails, the InstallAnywhere log files are placed in the following directories:

- UNIX™, Linux™: in your home directory.
- Windows™: on the Windows™ Desktop.

Debug and tracing

You can set the level of tracing detail that is used and the location of the debug file by specifying the `DDEBUG_LEVEL` and `DDEBUG_FILE` options on the installation command. By default tracing is turned off. For more information, see [Client products installation command syntax on page 45](#).

Return codes (UNIX™, Linux™)

You can check the InstallAnywhere return code, also known as the *exit code*, after running the installation application to learn more about the installation status and possible error conditions. To retrieve the return code, run the following command after the installation application finishes:

```
echo $?
```

For information about the meanings of the InstallAnywhere return codes, see the documentation at the Flexera Software website: <http://www.flexerasoftware.com>.

Installing HCL OneDB™ Client Products

The *HCL OneDB™ Client Products Installation Guide* explains how to install HCL OneDB™ Client Software Development Kit (Client SDK).

This information is for database administrators or software engineers who install HCL® OneDB® client products. You must be familiar with the operating procedures of your computer and with your operating systems before installing these products. These topics explain how to install Client SDK as individual products using stand-alone client installation applications. Alternatively, you can install Client SDK as part of the HCL OneDB™ software bundle installation media. The software bundle includes both the HCL OneDB™ database server and client products.

Index

Special Characters

- .NET Provider 35
- /etc/hosts file
 - multiple residency 29
- /etc/services file
 - multiple residency 29
- /etc/termcap file 24
- /etc/terminfo directory 24

A

- AAO
 - see auditing analysis officer 12
- administrative access 8
 - group and user informix 9
- Architecture 4
- auditing analysis officer (AAO) role 12

B

- Backups
 - and multiple residency 27
- backups, preparing for multiple residency 29

C

- CC8BITLEVEL environment variable 48
- Client products
 - uninstalling 53
- Client SDK
 - Environment Variables 48
 - uninstalling 53, 54
- CLIENT_LOCALE environment variable 48
 - setting 24
- commands
 - onedb_install 14
 - uninstallids 32
 - uninstalling client products 53, 54
 - uninstalling Client SDK 53, 54
 - uninstalling HCL OneDB
 - Connect 53, 54
 - uninstallserver 32, 33
 - commandsininstallconnect
 - installclientsdk 45
- Components 4
- configuration file 23
- configuration parameters 23
 - MIRROROFFSET 29
- configuring
 - role separation 12
- Configuring
 - HCL OneDB products 50
- CONNECT_RETRIES environment variable 48
- Connection Manager 35
- Connections 48
 - connectivity 24
 - configuring for multiple residency 29
- Create a server
 - installation 13
- Creating
 - database server instances 27
- creating a server 22
- custom installation 11, 13, 14, 40

D

- Database server
 - starting 25

- stopping 25
- database servers
 - running multiple 29
- Database servers
 - creating new instances 27
 - identifying 48
- database system security officer (DBSSO) role 12
- DATE data type 48
- DATETIME data type 48
- DB_LOCALE environment variable 48
- DBANSIWARN environment variable 48
- DBFLTMASK environment variable 48
- DBLANG environment variable
 - setting 24
- DBMONEY environment variable 48
- DBMONEYSCALE environment variable 48
- DBPATH environment variable 48
- DBSERVERNAME parameter
 - multiple residency 29
- DBSSO
 - see database system security officer 12
- DBTEMP environment variable 48
- DELIMITIDENT environment variable 48
- Delimited identifiers 48
- deployment wizard 13
- directories
 - installation 10, 41
- disk space
 - for multiple residency 29
- Disk space 37
- disk space requirements 5, 5, 6, 37, 38
- drsoctcp 52
- DYLD_LIBRARY_PATH environment variable, setting 24

E

- environment variables
 - CLIENT_LOCALE 24
 - DBLANG 24
 - DYLD_LIBRARY_PATH 24
 - environment variables
 - LD_LIBRARY_PATH 24
 - GL_USEGLU 24
 - LIBPATH 24
 - LIBPATH environment variable (AIX), setting 24
 - ONEDB_SQLHOSTS 24
 - ONEDB_SERVER
 - setting 24
 - shared library path 24
 - SHLIB_PATH 24
 - TERMCAP 24
- Environment variables
 - CC8BITLEVEL 48
 - CLIENT_LOCALE 48
 - CONNECT_RETRIES 48
 - CONNECT_TIMEOUT environment variable 48
 - Database locale, setting language features 48
 - Database servers
 - default 48
 - DB_LOCALE 48
 - DBANSIWARN 48
 - DBFLTMASK 48
 - DBLANG environment variable 48
 - DBMONEY 48

- DBMONEYSCALE 48
- DBPATH 48
- DBTEMP 48
- DELIMITIDENT 48
- Environment variables
 - CONNECT_TIMEOUT 48
 - DBLANG 48
 - ONEDB_HOME 48
 - ONEDB_SERVER 48
 - OPTMSG 48
 - OPTOFC 48
 - TMP 48
- ESQLMF 48
- FET_BUF_SIZE 48
- GL_DATE 48
- GL_DATETIME 48
- IFX_FLAT_USCQ 48
- MONEY data type 48
- NCHAR data type 48
- NVARCHAR data type 48
- ONEDB_SQLHOSTS 48
- ONEDB_SQLHOSTS environment variable 48
- ONEDB_SERVER environment variable 48
- OPTMSG environment variable 48
- OPTOFC environment variable 48
- TMP environment variable 48
- Environment Variables
 - Client SDK 48
- ESQL/C 35
- ESQLMF environment variable 48
- exit codes 21, 56

F

- Failover
 - Microsoft cluster 26
- FET_BUF_SIZE environment variable 48
- files 43
 - error 21
 - for connectivity 24
 - installation log 21
 - installation tracing 21
 - response 15, 16, 17, 43
 - uninstallids.exe 33

G

- GL_DATE environment variable 48
- GL_DATETIME environment variable 48
- GL_USEGLU environment variable 24
- GLS. 35
 - group informix 8, 9
 - groupadd utility 9, 9

H

- HCL OneDB
 - Connect
 - definition 35
 - uninstalling 53, 54
 - hosts file 24

I

- IFX_FLAT_USCQ environment variable 48
- ILogin utility 35
- informix group 8
- informix user 8
- Informix-Admin group 12
 - using Server Instance Manager 27
- Install medium 37
- installation

- commands 18
- custom 13, 14
- debugging 21
- directory 10, 41
- disk space 5, 5, 6, 37, 38
- error 21
- log files 21
- silent 15, 16, 17, 43
- typical 14
- typical or custom 11, 40
- unattended 15, 43
- Installation
 - create a server 13
 - preparing for 3, 35
- installclientsdk command 45
- installconnect command 45
- installing
 - by extracting media with script 44
- interactive installation 11, 40

J

- Java runtime environment requirements 5, 5, 6, 37, 38
- JRE requirements 5, 5, 6, 37, 38

L

- LD_LIBRARY_PATH environment variable, setting 24
- legacy installation support 44
- log files
 - deploy 21, 56
 - error 21, 56
 - iad_act 56
 - iad_act.log 21
 - install 21, 56
 - output 21, 56

M

- Memory requirements, and multiple residency 27
- Message file 48
- Message log, for multiple residency 27
- Microsoft Cluster Server 26
- Microsoft Failover Server 26
- Mirroring, and multiple residency 27
- MIRROROFFSET configuration parameter
 - multiple residency 29
- MIRRORPATH parameter
 - multiple residency 29
- MSGPATH parameter
 - multiple residency 29
- Multiple residency 27
 - and backups 27
 - and mirroring 27
 - and multiple binary files, warning 27
 - memory requirements 27
 - message log for 27
 - onconfig file 27
 - planning for 27
 - storage-space requirements 27
- multiresident database servers 29

N

- non-interactive installation 11, 40
- non-root install 8
- non-root installation 10, 41, 41

O

- Object Interface for C++ 35
- ODBC-MTS 35
- OLE DB Provider 35, 35
- onconfig file, and multiple residency 27

- onconfig parameters 23
- onconfig.std file 23
- ONEDB_SQLHOSTS environment variable, setting 24
- ONEDB_HOME 10, 41
 - lib directory 24
- onedb_install command 14, 18
- ONEDB_SERVER environment variable
 - setting 24
- oninit command 26
- onipcnp 52
- onipcshm 52
- onipcstr 52
- onmode -s command 26
- onsecurity utility 10
- onsocimc 52
- onsoctcp 52
- ontliimc 52
- ontlitcp 52
- operating system requirements 5, 5, 6, 37, 38

P

- passwd utility 9
- Planning
 - installation 3, 35
- Preparing
 - for installation 3, 35
- Privileges
 - local 27
 - Server Instance Manager 27
- Processor, storing temporary files 48
- Products 4

R

- redistributing
 - HCL OneDB products 44
- removing features 32
- response file 15, 16, 17, 43
- return codes 21, 56
- role separation
 - configuring 12
- roles 12
- ROOTOFFSET configuration parameter
 - multiple residency 29
- ROOTPATH parameter
 - multiple residency 29

S

- s 44
- scripts
 - for installing by extracting media 44
- server configuration 22
- server instance
 - creating during installation 14
 - non-root installation 14
- Server Instance Manager
 - command-line options 28
 - DRDA (Distributed Relational Database Architecture) 28
 - Informix-Admin Group 27
 - privileges 27
 - starting 27
- SERVERNUM configuration parameter
 - multiple residency 29
- Services application 26
- services file 24
- Setnet32 utility
 - product configuration 50
- shared-library path environment variable
 - setting 24

- SHLIB_PATH environment variable (HP-UX)
 - setting 24
- shutdown script
 - multiple residency 29
- silent installation 15, 16, 17, 43
- sqlhosts file 24
 - multiple residency 29
- Starting the server 26, 26
- starts command 26
- startup script
 - multiple residency 29
- Stopping the server 26, 26
- Storage-space requirements, and multiple residency 27
- System requirements 37
- system startup script
 - multiple residency 29

T

- TERMCAP environment variable
 - setting 24
- typical installation 11, 14, 40

U

- unattended installation 15
 - see also silent installation 43
- Uninstall
 - client products 53
- uninstallids command 32
- uninstallids.exe file 33
- uninstalling 32, 32, 33
 - database server 32, 33
 - HCL OneDB software bundle 32, 33
- Uninstalling client products
 - silent 54
- uninstallserver command 32, 33
- user informix 9
- User informix 8
- useradd utility 9
- utilities
 - groupadd 9
 - passwd 9
 - useradd 9

W

- wire listener 22