

**HCL OneDB 2.0.1**

**OneDB Embeddability Guide**



# Contents

- Chapter 1. Embedding HCL OneDB™ ..... 3**
  - Overview of embedding HCL OneDB™ products..... 3
  - Customize the size of HCL OneDB™ ..... 3
  - Customize a silent installation..... 3
  - Easy deployment..... 4
  - Synchronize connectivity information between servers and clients on Windows™ ..... 4
  - Automated administration..... 5
  - Easy configuration in embedded environments..... 5
  - Specify the HCL OneDB™ environment in the onconfig file..... 6
  - Maintain HCL OneDB™ availability..... 7
- HCL OneDB™ embeddability deployment tutorial..... 21
  - Introducing the HCL OneDB™ deployment assistant..... 22
  - Introducing the HCL OneDB™ deployment utility..... 22
  - Automating deployments of reduced-footprint HCL OneDB™ on Linux™ ..... 25
  - Automating deployments of reduced-footprint HCL OneDB™ on Windows™ ..... 28
- Manage memory in embedded environments..... 32
  - Optimize storage..... 32
  - Manage message logs in an embedded environment..... 33
  - Reserve memory for critical activities in embedded environments..... 37
  - Maintain a targeted amount of memory in embedded applications..... 37
- Index..... 39**

# Chapter 1. Embedding HCL OneDB™

The *HCL OneDB™ Embeddability Guide* describes how to embed HCL OneDB™ in your applications.

This information is intended for application programmers.

## Overview of embedding HCL OneDB™ products

You can embed HCL OneDB™ products in your applications by taking advantage of customizable deployment options and configurable administration features. You can use the embeddability features in combination with other product features for installation footprint reduction and high-availability clustering to achieve a quicker and easier end-to-end deployment solution.

### Customize the size of HCL OneDB™

If your application runs in a low-memory environment, you might need to minimize the size of HCL OneDB™.

You can customize the size of HCL OneDB™ during installation and during snapshot creation.

You can use the Deployment wizard during installation to exclude parts of the HCL OneDB™ product that your application does not need. The installation program ensures that all dependent components are automatically selected. The installation program shows you what the size of the product will be after installation. The minimum size of the HCL OneDB™ installation is approximately 100 MB.

You can use the Deployment Assistant to exclude parts of the product from the snapshot of an existing HCL OneDB™ instance that you intend to deploy embedded in your application. The Deployment Assistant does not enforce dependencies. You can also choose which HCL OneDB™ client products and which dbspaces being used by the HCL OneDB™ instance to include in the snapshot.

---

#### Related information

[Exclude database server components on page](#)

[The deployment utility on page 7](#)

### Customize a silent installation

You can create a silent installation script, customized for your requirements, that you can use multiple times.

To create a custom silent installation script, you install HCL OneDB™ using the GUI- or console-mode to capture installation settings that you plan to reuse in multiple locations (often different host environments altogether, such as multiple computers). The installation settings are captured in a *response file*, which serves as a template for the installation setup to be deployed. When you run the silent installation command for the target location, the response file is specified on the command line so that same installation settings are replicated by running the script.

To expedite the time for using an embedded solution, you can deploy a silent installation as a non-root installation, which does not require any system administration privileges or user and group **informix** accounts. For details, see information

about non-root installation in the *HCL OneDB™ Installation Guide* *HCL OneDB™ Installation Guide for UNIX™, Linux™, and Mac OS X*.

---

#### Related information

[Installation owner on page](#)

[Running a silent installation on page](#)

## Easy deployment

You can use the deployment tools to quickly and easily deploy HCL® OneDB® to multiple locations.

You can take a snapshot of a fully configured HCL OneDB™ instance and its databases with the deployment assistant. You can then deploy that snapshot with the deployment utility to multiple target computers with one command.

The HCL OneDB™ embeddability tutorial provides step-by-step methods for using the deployment tools and writing a silent installation script. Specifically, the tutorial walks you through archiving of an installed database server instance with its associated data spaces on a template computer and deploying the snapshot on a different computer. A script-based method, such as described in the tutorial, can maximize the embeddability benefits of the deployment tools.

---

#### Related information

[Creating a snapshot with the deployment assistant on page 9](#)

[Creating a snapshot for deployment on Linux on page 25](#)

## Synchronize connectivity information between servers and clients on Windows™

On the Windows™ operating system, you must configure connectivity information for HCL OneDB™ Servers by using the `sqlhosts` file.

The `sqlhosts` file is installed in `%ONEDB_HOME%\etc\sqlhosts.%ONEDB_SERVER%`. The `sqlhosts` file on the Windows™ operating system uses the same format as the `sqlhosts` file on UNIX™ operating systems. The **sync\_registry** task automatically converts the connection information between the `sqlhosts` file format and the Windows™ registry format.

When maintaining several client machines, you can update their registries by using the **syncsqlhosts** utility. For example, if a DRDA® connection is added to a server, all client machines can use this utility to update their registries as well.

---

#### Related reference

[The syncsqlhosts utility on page 19](#)

## Automated administration

You can automate database server administration to minimize or eliminate the need for a database administrator after deployment of the database server instance as an embedded solution.

You can configure autonomic features that are included with HCL OneDB™:

- Autonomic configuration parameters control how the server does self-tuning and recovery. Many automatic tuning options are enabled by default. You can set other tuning options.
- Scheduler tasks perform maintenance tasks at configurable intervals for configurable thresholds. For example, the Auto Update Statistics (AUS) maintenance system updates stale table statistics.

If you create a demonstration server during installation, many automatic features are configured based on the information that you provide during installation.

You can monitor HCL OneDB™:

- The **sysadmin:ph\_alert** table contains server issues, ranked by severity. The issues are the results of built-in Scheduler sensors and event alarms.

You can create customized responses and corrective actions to server issues:

- You can create Scheduler tasks to respond to alerts, event alarms, and other situations.
- You can use the SQL administration API in Scheduler tasks and other SQL statements to perform administration tasks remotely.

---

### Related information

[Automatic performance tuning on page](#)

[The Scheduler on page](#)

[Remote administration with the SQL administration API on page](#)

[The ph\\_alert Table on page](#)

[onconfig Portal: Configuration parameters by functional category on page](#)

[Database server configuration during installation on page](#)

## Easy configuration in embedded environments

You can configure and tune the database server in embedded environments, regardless of the workload or information you have about resources.

Using one configuration parameter, the AUTO\_TUNE configuration parameter, you can enable or disable all automatic tuning parameters. You can also replace multiple automatic tuning configuration parameters in your configuration file with the AUTO\_TUNE configuration parameter.

You can export a snapshot of your current configuration parameters to a file. You can archive this file, use it as a configuration file, or import it to another running instance. Importing a configuration file is a convenient way of dynamically modifying multiple tunable parameters at one time. If the file contains configuration parameters that cannot be tuned dynamically, the values of those parameters are ignored.

---

**Related information**

[Database server configuration on page](#)

[onmode -wf, -wm: Dynamically change certain configuration parameters on page](#)

[onmode -we: Export a file that contains current configuration parameters on page](#)

[onmode -wi: Import a configuration parameter file on page](#)

[import config argument: Import configuration parameter values \(SQL administration API\) on page](#)

[export config argument: Export configuration parameter values \(SQL administration API\) on page](#)

[AUTO\\_TUNE configuration parameter on page](#)

[Specify the HCL OneDB environment in the onconfig file on page 6](#)

## Specify the HCL OneDB™ environment in the onconfig file

You can set local environment variables in the `onconfig` file for the HCL OneDB™ instance. These settings, which are independent of the global or system environment variable settings on the computer, can be used by the `oninit`, `oncheck`, `onclean`, `onlog`, `onmode`, `onparams`, `onspaces`, and `onstat` utilities in embedded environments.

When you use the `-FILE` option with one of the specified utilities, you can:

- Start the database server from a remote machine without having to create specific environment files and wrappers for each instance or without having to set the environment in a shell.
- Define information for the server to use automatically instead of setting environment variables for an instance before you start the database server.
- Run the HCL OneDB™ utility without having to set your environment explicitly in the shell before you run the utility.
- Have a single point to specify environment variables for instances in order to further embed the underlying implementation and installation of the database server.

---

**Related information**

[Setting local environment variables for utilities on page](#)

[The `-FILE` option on page](#)

[Easy configuration in embedded environments on page 5](#)

## Maintain HCL OneDB™ availability

You can ensure that your applications can always access HCL OneDB™ even if a single server fails by configuring a high-availability solution.

Sometimes HCL OneDB™-embedded applications are deployed to environments where a DBA might not be immediately available. You can configure these applications to switch to another server if the original database server has a problem or is down for maintenance.

You can choose between different high-availability configurations depending on your needs:

- A high-availability cluster consists of a primary server and one or more secondary servers that are copies of the primary server. Secondary servers can be shared-disk, local, or remote. You can configure automatic connectivity based on available servers.
- Enterprise Replication allows a great deal of flexibility in how the participating servers are organized and whether all or a subset of the data is replicated.

---

### Related information

[IBM Informix Enterprise Replication technical overview on page](#)

## Deploying HCL OneDB™

The product supports embeddability tools that enable lightweight, faster deployment of the database server and related client programs.

### About this task

Use the command-line deployment utility to deploy a snapshot of HCL OneDB™ and create a new instance as a quick alternative to traditional installation, especially for deployments to many computers. You can configure the snapshot so that the instance is ready to meet your requirements immediately after deployment.

You can customize an RPM Package Manager image of the HCL OneDB™ installation and then distribute the image to other locations on supported Linux™ platforms.

---

### Related information

[Introducing the HCL OneDB deployment utility on page 22](#)

## The deployment utility

The HCL® OneDB® deployment utility can deploy snapshots of pre-configured HCL OneDB™ instances (with or without data) on one or more computers.

A snapshot is an image of the HCL OneDB™ database server that includes the installation directory, configuration settings, and any data spaces 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 customization of the snapshot.

Deploying a snapshot is quicker than installing HCL OneDB™ with the silent installation option. When you deploy a snapshot on a computer, you can replace the snapshot or remove it by using the deployment utility.

The following list describes some scenarios for which the deployment utility can be useful:

- You want to deploy a particular database server configuration on multiple computers. You can tune only one instance as the template instance, and then use the utility to deploy it on to other computers in silent mode.
- You want to clone an instance on the same computer or set up multiple instances quickly.
- You want to upgrade multiple instances to a different fix pack or version level to take advantage of newer product enhancements but this requires tuning certain configuration parameters or environment variables. You can tune the template instance, and then use the utility to upgrade other instances rapidly.
- You are embedding the HCL OneDB™ application on multiple computers and want to reduce application installation and setup time. You can specify the installation location and a single path for all application files including the database server files during deployment to avoid waiting for data loading and database server initialization.

---

#### Related information

[Customize the size of HCL OneDB on page 3](#)

## Planning for deploying HCL OneDB™

Before you deploy HCL OneDB™, you should decide what you want to include in your deployment and how to configure the target server.

On the source server, you create a snapshot of the database server and, optionally, of the data in the database. Before you create a snapshot, decide on the properties of the snapshot. When you run the ifxdeployassistant utility in GUI mode, you can choose the following properties of the snapshot:

- The server features and components to include. You can choose which features and components to include in the snapshot. By default, all features and components are included in the snapshot.
- The data to include. You can create snapshots of your dbspaces. When you deploy them to the target computer, the dbspaces are already initialized. By default, no dbspaces are included in the snapshot.
- The archive format. You can choose one of three tar archive formats with or without compression, or a zip archive with compression. By default, the format is a tar archive with Gzip compression.

On the target server, you deploy the snapshot of the server and any dbspaces and configure the environment. Before you deploy the snapshot, decide on the properties of the target server:

- Where to put the server. If there is no HCL OneDB™ installation on the machine, you can deploy a new installation using an archive of the ONEDB\_HOME directory from your source machine. You can deploy with the following command:

```
ifxdeploy -INSTALL -i ONEDB_HOME --snapshot archive_file
```

The *archive\_file* is uncompressed in the directory specified by the -i parameter or the ONEDB\_HOME environment variable.

- Where to put the data. If you create snapshots of the data, and all the dbspaces are in a single file and are deployed to a single directory, then you can use the `-image_archive archive_file,dest_directory` option for ifxdeploy to create the directory and uncompress the snapshot for you. If your dbspaces will be in multiple directories then you must create directories for the dbspaces yourself, set the proper ownership and permissions, and extract the snapshot contents into the directories. By default, the ifxdeploy utility uses the same directories as on the source server. You can instruct the server to use new directories with the `--relocate` option in the ifxdeploy command.
- The configuration of the environment. You must set the ONEDB\_HOME, ONEDB\_SERVER, and password values when you run the ifxdeploy command. You can set other server properties.
- The configuration of the server. You can customize the configuration parameters for the target server. By default, an `onconfig` file is created based on the `onconfig.std` file. You can specify a different `onconfig` file with the `-o` option of the ifxdeploy command.

#### Related information

[Creating a snapshot with the deployment assistant on page 9](#)

[HCL OneDB embeddability deployment tutorial on page 21](#)

## Creating a snapshot with the deployment assistant

Use the deployment assistant to create a snapshot of a running HCL OneDB™ database server instance without shutting it down. You can reduce the snapshot footprint by selecting which features in the template instance to omit. You can include or exclude associated data spaces.

### Before you begin

Before you create a snapshot, you must meet the following prerequisites:

- A supported Java™ runtime environment must be on the system and must be set in the PATH environment variable.
- A Java™ runtime environment, version 1.6 or later must be on the system and must be in the PATH setting. The runtime environment must match the bit level of the HCL OneDB™ installation. You can use the runtime environment that is installed with HCL OneDB™: `ONEDB_HOME/extend/krakatoa/jre` on UNIX™ and Linux™ and `%ONEDB_HOME%\extend\krakatoa\jre` on Windows™.
- You are able to run the deployment assistant on the computer that hosts the HCL OneDB™ instance to be used for the snapshot. (The deployment assistant cannot capture a snapshot from a remote computer.)
- You must have the CONNECT privilege on the **sysadmin** database to include any data spaces in the snapshot.

- Data spaces that you want to include in the snapshot must be stored in chunks that use buffered files (also known as cooked files).
- There must be sufficient disk space to save the snapshot in a file. You can store the snapshot in a compressed archive file.

### About this task

The deployment assistant has two modes: the GUI mode and the command-line mode. You can only omit components and features by using the GUI mode.

To create a snapshot with the deployment assistant in GUI mode:

1. Run the `ifxdeployassist` command without options from the `%ONEDB_HOME%\bin` (Windows™) or `$ONEDB_HOME/bin` (UNIX™ or Linux™) directory to start the deployment assistant GUI.
2. If the deployment assistant does not detect the instance from which you want to create a snapshot, enter the connection information that pertains to the instance that you want to use as a template.
3. Specify the path, file name, and archive format for the snapshot that you are creating.
4. Select components and features that you want to include in or exclude from the snapshot.



**Important:** The deployment assistant does not enforce functional interdependencies between product components and features.

5. If there are data spaces associated with the instance, select whether or not to include the data spaces in the snapshot.  
The deployment assistant GUI displays each data space, along with the chunk locations in each data space.
6. Confirm that the snapshot settings are correct.

### Results

To create a snapshot with the deployment assistant from the command line:

Run the `ifxdeployassist -c` command along with any other appropriate command options.

If data spaces are included in the deployed snapshot, the deployment assistant summary displays the full path name of the data spaces package.

---

#### Related information

[Easy deployment on page 4](#)

[Planning for deploying HCL OneDB on page 8](#)

[Introducing the HCL OneDB deployment assistant on page 22](#)

## The ifxdeployassist command

Use the ifxdeployassist command to create and customize a snapshot. If you do not pass any options on the command line, the deployment assistant runs in GUI mode.

### Syntax

```
ifxdeployassist [ -aarchive_format ][ -fpath_name ][ -iconnectivity_information ][ -c ] { [ -d [ -n ] ] } [ -h ] [ -v ]
```

### Command options

**Table 1. Options for the ifxdeployassist command**

Opt ion	Purpose	Key Considerations
-a arc hiv e_f or mat	Specifies the archive format and compression type to save the snapshot.  The following values can be used in the place of <i>archive_format</i> :  <ul style="list-style-type: none"> <li>BZIP2 (tar archive with BZip2 compression)</li> <li>GZIP (tar archive with GZIP compression)</li> <li>TAR (tar archive with no compression)</li> <li>ZIP (zip archive with compression)</li> </ul>	If you do not pass the -a option in the command, the snapshot is saved as a tar archive with Gzip compression. If you use the standard Gzip format to create snapshots of data spaces, each data chunk must be less than or equal to 4 GB.
-f pat h_n ame	Specifies the path and file name of the snapshot to be created.	If you do not specify the -f option, the default file name of the server snapshot is:  <i>server_name_yymmdd-hhmm.tar.gz</i>  The <i>server_name</i> is the name of the server. The <i>yymmdd-hhmm</i> is the time the snapshot is taken.  The default file name of the data snapshot is:  <i>server_name_yymmdd-hhmm_db.tar.gz</i>  The archive and compression selection set by the value in the -a option overrides any archive and compression settings indicated in the file name suffix set with the -f option. For example, if a command contains the following: <code>ifxdeployassist -a TAR -f C:\HCL OneDB™\server.zip</code> then the snapshot will be saved as a tar archive with no compression rather than as a zip archive with compression.

**Table 1. Options for the ifxdeployassist command (continued)**

Opt ion	Purpose	Key Considerations
<code>-i</code> <code>con</code> <code>nectiv</code> <code>ity</code> <code>_in</code> <code>for</code> <code>mat</code> <code>ion</code>	Specifies the connectivity information for the template instance from which to create a snapshot. If you do not use the <code>-i</code> option, the deployment assistant detects the connectivity information from the current environment and uses the same connectivity settings.  In the place of <i>connectivity_information</i> , indicate specific connectivity settings for the snapshot to use after it is deployed. Set values for the following variables, separated by colons without spaces: <ul style="list-style-type: none"> <li>• name of the database server instance</li> <li>• name of the template computer for which the database server instance is configured</li> <li>• service port number</li> <li>• protocol for the connection</li> </ul>	The only valid values for the protocol information are <code>SQLI</code> and <code>DRDA</code> . If no protocol is specified, the command sets the connection to <code>SQLI</code> .  The connectivity information here is contained in the <code>sqlhosts</code> file.  <b>Example:</b> <code>ifxdeployassist -i hrapps:oscar:40:DRDA</code>
<code>-c</code>	Directs the deployment assistant to run in a command-line interface.	The <code>-c</code> option is required only if you are taking a snapshot of an instance that you do not want to customize (that is, accept all features of the template instance), but want to capture the snapshot in a scripting environment.
<code>-d</code>	Includes the data associated with the instance in the snapshot.	
<code>-n</code>	Captures a snapshot without the HCL OneDB™ database server instance installation.	You must run the <code>-d</code> option in the command in order to use the <code>-n</code> option.
<code>-h</code>	Displays the online help for the ifxdeployassist command.	
<code>-v</code>	Runs the ifxdeployassist command in verbose mode.	

## Usage

If you run the ifxdeployassist command without any options, the deployment assistant starts in GUI mode instead of as a command-line interface.

## Examples of script-based usage of the deployment assistant

The examples of snapshot creation and modification below are based on a Windows™ environment in which:

- An installation of the HCL OneDB™ database server exists in `C:\informix`.
- The `ifxdeployassist` command is run in `C:\informix\bin`.
- The **ONEDB\_SERVER** environment variable is set to `informix1`.
- The template instance has data associated with it.

### Snapshot containing only the data space for the server

The following command creates a snapshot of the server only at `C:\informix\bin\informix1.tar.gz`.

```
ifxdeployassist -c
```

### Snapshot containing the server and data, with specified archive type and full path name, from a template instance outside the current environment variable setting

The following command creates a snapshot of the database server at `C:\temp\snapshot_1.zip` and of the associated data at `C:\temp\snapshot_1_db.zip`. The value `informix2:localhost:9090` directs the deployment assistant to take a snapshot of the `informix2` instance, which is configured for `localhost`, and to set the service port to `9090`. Because neither `DRDA@` nor `SQLI` is specified after `9090`, the connectivity port of the snapshot is set to use the `SQLI` protocol.

```
ifxdeployassist -c -d -i informix2:localhost:9090
-a zip -f C:\temp\snapshot_1
```

### Snapshot in specified archive type capturing only the data associated with the instance, running in verbose mode

The following command creates a snapshot of data at `C:\Informix\bin\informix2_db.tar`. The template instance, host computer name, and connectivity information used are the same as described in the previous example.

```
ifxdeployassist -cvndi informix2:localhost:9090 -a tar
```

## Creating a snapshot for deployment manually

Create a snapshot of HCL OneDB™ that you can use with the deployment utility to place pre-configured instances on multiple computers.

### Before you begin

Before you create a snapshot, you must meet the following prerequisites:

- **Windows™**: Windows™ administrator privileges on the computer.
- **Linux™ and UNIX™**: User `informix` or root privileges on the computer.
- Sufficient disk space to save the snapshot in a file. If space is limited, you might want to store the snapshot in a compressed file.

### About this task

To create a snapshot, complete the following steps on the computer where you installed the HCL OneDB™ instance:

1. Shut down the instance in a consistent state with `onmode -kuy`.
2. Create a snapshot of the following items:
  - HCL OneDB™ installation directory

 **Tip:** Ensure you include the `IDSFILES.txt` file, which is in the installation directory. That file is required if you want to use the deployment utility to remove the snapshot from the target computer after you deploy it.

- Configuration settings
- Optional: Data spaces associated with the instance

 **Tip:** Store the components of the snapshot in a compressed file if you want to save space. If you compress the snapshot in a BZIP2, GZIP, TAR, or ZIP format, you can use the `-file` option with the deployment utility to extract the snapshot instead of specifying a customized decompression command.

3. **Optional:** After you create the snapshot, you can restart the instance.

## Deploying a snapshot with the deployment utility

Deploy a snapshot of an instance by using the `ifxdeploy` utility on the target computer.

### Before you begin

Before you deploy a snapshot, you must meet the following prerequisites:

- **Windows™:** Windows™ Administrator privileges on the target computer.
- **Windows™:** The required Visual C++ runtime libraries must exist on the target computer. You can install these libraries by one of the following methods:
  - Copy the `%ONEDB_HOME%\bin\vcredist.exe` file to the target computer and run the executable file.
  - Run the following command on the target computer:

```
vcredist.exe /q:a /c:"msiexec /i vcredist.msi /qn"
```

- **Linux™: and UNIX™:** Root privileges on the target computer, if you want the instance to be owned by user **informix**. Alternatively, deploy the snapshot as a non-root user.
- The target computer has sufficient disk space for the snapshot. You must have the same amount of space as was used on the template computer. The space required depends on what you included in the snapshot, such as data, extra files, or other applications.
- The target computer must be in the same operating system family as the source computer.

### About this task

This procedure describes how to deploy that snapshot with a configuration file. Alternatively, you can use command-line options to specify the same information that is contained in the configuration file.

To deploy the snapshot on the target computer:

1. Create the directory for the server and save the following files in that directory:

**Choose from:**

- The server snapshot
- The ifxdeploy utility file

2. If you have a data snapshot (valid formats are `.zip` and `.tar`), you can instruct the ifxdeploy utility to automatically deploy the snapshot with the `--image_archive archive_file, dest_directory` option. The destination directory must not exist. If you use this option you must also specify the `--rootpath` option to configure the current root chunk of the instance. If the data spaces on the source computer are not all in the same directory, include the `--relocate` option with the correct directory names for each data space.

3. Include the following ifxdeploy command options and any other options you need:

**Choose from:**

- `-i ONEDB_HOME`
- `-s ONEDB_SERVER`
- `--ixpass password`

4. Run the ifxdeploy command with the appropriate options.

5. **Windows™:** Manually add the HCL OneDB™ .NET drivers to the Global Assembly Cache (GAC).

## The ifxdeploy command: The deployment utility

Use the ifxdeploy command to deploy a snapshot or remove a snapshot that you already deployed.

### Syntax

#### Installation command

```
ifxdeploy -INSTALL -i ONEDB_HOME --snapshot snapshot_file [ --verbose ] [ --nonroot ] [ --silent ]
```

#### Create instance command

```
ifxdeploy -INSTANCE create -i ONEDB_HOME -s ONEDB_SERVER [ -o ONCONFIG ] --ixpass password [ --sqlhosts sqlhosts_file ] [ --dbalias name = alias , protocol = protocol , port = port ] [ --servername server_number ] [ --image_archive archive_file , dest_directory ] [ --rootpath rootpath ] [ --relocate { path | old_path = new_path | old_path , old_offset = new_path , new_offset } ] [ --start number ] [ --verbose ] [ --nonroot ] [ --silent ]
```

#### Uninstallation command

```
ifxdeploy -UNINSTALL -i ONEDB_HOME [ --verbose ] [ --deluser ] [ --nonroot ] [ --silent ]
```

## Command options

Table 2. The ifxdeploy command options

Element	Purpose	Key considerations
--deluser	 <b>Windows:</b> Removes user <b>informix</b> and the <b>Informix-Admin</b> group.   <b>Linux and UNIX:</b> Removes user <b>informix</b> and group <b>informix</b> or the non-root user who owns the instance.	The --delifx option functions when there is only one HCL OneDB™ installation on the computer.
--dbalias name = <i>alias</i> , protocol= <i>protocol</i> , port= <i>port</i>	Specifies new database server aliases and related SQLHOSTS connectivity settings for the deployed instance.	<p>Each <i>alias</i> name must be unique and conform to server alias name rules. See <a href="#">DBSERVERALIAS configuration parameter on page</a> .</p> <p>The <i>protocol</i> must be a valid connection type for the operating system. The <i>port</i> must be a valid port number. See <a href="#">sqlhosts connectivity information on page</a> .</p>
-i <i>ONEDB_HOME</i>	Path for the deployed instance. Must be set here or as environment variable (no default path is provided).	
--ifxpass <i>password</i>	Specifies the password of the user used to create the HCL OneDB™ service.	Specify the password for user <b>informix</b> or the non-root user on the target computer. If you specify a password and the <b>informix</b> user does not exist on the target computer, the <b>informix</b> user is created with the specified password.
--image_archive <i>archive_file</i> , <i>target_directory</i>	Specifies the path of the snapshot archive file and the target directory on the target computer. The directory is created and the snapshot is uncompressed.	<p>Valid formats for the <i>archive_file</i> are <code>.zip</code> and <code>.tar</code>.</p> <p>If your dbspaces will be in multiple directories, do not use this option. You must create directories for the</p>

Table 2. The `ifxdeploy` command options (continued)

Element	Purpose	Key considerations
--nonroot	Specifies that the instance is owned by the non-root user who runs the command.	dbspaces yourself, set the proper ownership and permissions, and extract the snapshot contents into the directories.  UNIX and Linux only.
-o <i>ONCONFIG</i>	The <code>onconfig</code> file name. If none is specified here and no <b>ONCONFIG</b> environment variable is set, a new file is created from the <code>onconfig.std</code> file.	
--relocate <i>path</i>	Specifies new parent directory of chunks in the deployed instance. You must have the appropriate ownership and permissions to modify the directories.	You can specify a single new parent directory or map multiple separate chunk paths to different locations. In addition, you can substitute offsets of the old paths with new values if you are indicating multiple new parent directories.  <div data-bbox="1084 1094 1435 1335">  <b>Windows:</b> If the parent directories have spaces in the paths, place the string of paths after the <code>--relocate</code> option in double quotation marks. </div> <div data-bbox="1084 1398 1435 1671">  <b>Linux and UNIX:</b> If multiple parent directories are specified and have spaces in the paths, place the string of paths after the <code>--relocate</code> option in single quotation marks. </div>
--rootpath <i>rootpath</i>	Indicates the location of the root dbspace.	<div data-bbox="1084 1755 1435 1827">  <b>Windows:</b> Default path:<code>drive:\ifmxdata\s</code> </div>

**Table 2. The ifxdeploy command options (continued)**

Element	Purpose	Key considerations
<code>-s ONEDB_SERVER</code>	Primary database server name. Must be set either here or in the <b>ONEDB_SERVER</b> environment variable before deployment (no default value is provided).	 <code>erver_name\rootdbs_data.000</code>   <b>Linux and UNIX:</b> Default path: <code>\$ONEDB_HOME/server_name/rootdbs</code>
<code>--servername server_number</code>	Specifies the server number of the instance.	The number must be an integer from 0 to 255. The default is 0.
<code>--silent</code>	Directs the utility to run in silent mode.	When you use this option, messages do not display on the screen but are written in a log file. If you use the <code>-silent</code> option and <code>-verbose</code> option together, you must also include the <code>-l</code> argument.
<code>--snapshot snapshot_file</code>	Extracts a compressed snapshot of the file specified as <i>snapshot_file</i> .	 <b>Windows:</b> Indicate the location of the compressed image as an absolute path.
<code>--sqlhosts sqlhosts_file</code>	Specifies the name of the SQLHOSTS file.	Default is the value of the <b>ONEDB_HOMESQLHOSTS</b> environment variable, if set.
<code>--start number</code>	Starts the database server after the <i>number</i> of seconds.	Default is 600.
<code>-UNINSTALL</code>	Removes a snapshot that was originally deployed by the deployment utility.	
<code>--verbose</code>	Runs the command in verbose mode.	

## Usage

Before you use this command, create a snapshot and copy it to the computer where you want to deploy the snapshot. You must provide the location of the snapshot when you create the instance. See [Creating a snapshot for deployment manually on page 13](#) for more information.

Run this command on the computer where you want to deploy the snapshot. To deploy a database server, run the `ifxdeploy -INSTALL` command and then the `ifxdeploy -INSTANCE create` command. To remove an existing deployed database server, run the `ifxdeploy -UNINSTALL` command.

On Windows, you must run the command as an administrative user. On UNIX or Linux, you can run this command as the **root** user or as a non-root user. The user who runs the command is the owner of the database server instance.

If you included data spaces in your snapshot in compressed format, you must extract the data spaces before you run the `ifxdeploy` command.

## The syncsqlhosts utility

Use the **syncsqlhosts** utility to manually convert the connection information between the `sqlhosts` file format and the Windows™ registry format.

### Syntax

```
syncsqlhosts [ -sourcesqlhostvalue ] [ -targetsqlhostvalue ] [ -serversserverlist ]
```

### Command options

**Table 3. Options for the syncsqlhosts command**

Option	Purpose	Key considerations
-source <code>sqlhostvalue</code>	Specifies the source machine to copy connection information from. <code>sqlhostvalue</code> is either a \\machine name or an <code>sqlhosts</code> file.	If you do not specify a <b>-source</b> option, the local registry is used.
-target <code>sqlhostvalue</code>	Specifies the target machine to copy connection information to. <code>sqlhostvalue</code> is either a \\machine name or an <code>sqlhosts</code> file.	If you do not specify a <b>-target</b> option, the local registry is used.

**Table 3. Options for the syncsqlhosts command**

(continued)

Option	Purpose	Key considerations
- serv ers  serv erl ist	Specifies the servers to copy connection information from. <i>serverlist</i> is a list of HCL OneDB™ Server entries that are separated by spaces that are copied from the source to the target machine.	If you do not specify a <b>-servers</b> option, all of the defined connections are copied.

**Usage**

The following prerequisites must be met to run this command:

- The user must have the appropriate authorization to read or write files on the source and target system and registry.
- If the specified value for a parameter contains a space, you must enclose that value in double quotation marks ("").

**Example**

**Examples of syncsqlhosts utility usage**

In the following examples, you are issuing commands from a machine named machine1.

- To copy the connection information from the registry on machine2 to the registry on machine1:

```
syncsqlhosts -source \\machine2
```

- To copy the connection information from the registry on machine1 to the registry on machine3:

```
syncsqlhosts -source \\machine1 -target \\machine3
```

- To copy connection information from the registry on machine2 about servers ol\_server1 and ol\_server1test to the registry on machine1:

```
syncsqlhosts -source \\machine2 -servers ol_server1 ol_server1test
```

- To copy connection information from the registry on machine2 about servers ol\_server1 and ol\_server1test to a file on machine1:

```
syncsqlhosts -source \\machine2 -target c:\informix\etc\sqlhosts.ol_machine2  
-servers ol_machine1 ol_machine1test
```

**Related information**

[Synchronize connectivity information between servers and clients on Windows on page 4](#)

## Removing a snapshot with the deployment utility

Use the deployment utility to remove a snapshot that was originally deployed by the deployment utility. In addition, you can use this utility to remove the user who owns the installation from the operating system.

### Before you begin

**Windows™:** You must have Windows™ administrator privileges to remove the snapshot or remove the user and group objects.

**UNIX, Linux™:** You must have root privileges or be the non-root owner of the installation.

You must have the `IDSFILES.txt` file in the `etc` subdirectory of the installation path.

### About this task

To remove a snapshot:

Run the `ifxdeploy` command with the uninstallation options on the computer where the snapshot is deployed.

See [The ifxdeploy command: The deployment utility on page 15](#) for details about the command syntax.

### Example

The following command is an example of how to use the deployment utility to remove a snapshot and to display runtime status messages:

#### Windows™:

```
ifxdeploy -UNINSTALL -i C:\tmp\informix --verbose
```

#### UNIX, Linux™:

```
ifxdeploy -UNINSTALL -i /tmp/informix --verbose
```

The following is an example of how to uninstall a snapshot and to remove user **informix** and the administrative group (Windows) or the user who owns the installation (UNIX, Linux):

#### Windows™:

```
ifxdeploy -UNINSTALL -i C:\tmp\informix --deluser
```

#### UNIX, Linux™:

```
ifxdeploy -UNINSTALL -i /tmp/informix --deluser
```

## HCL OneDB™ embeddability deployment tutorial

The purpose of this tutorial is to provide steps and scripts needed for silent end-to-end deployment of HCL OneDB™ on Linux™ and Windows™ using the deployment assistant and the deployment utility.

The HCL OneDB™ Embeddability Toolkit is a logical collection of the following components:

- `ifxdeployassist`: the deployment assistant (DA).
- `ifxdeploy`: the deployment utility (DU).

- `ifxdeploy.conf`: the DU's configuration file.
- `ifx_silent_deploy`: an example script that automates silent deployment using DU.

The following tasks are covered in this tutorial for silent deployment of HCL OneDB™:

- Create a Snapshot for Deployment: Using the DA, archive an installed HCL OneDB™ server instance and its dbspaces on the template computer for future deployments.
- Silently Deploy HCL OneDB™ from the Snapshot: Using the DU, its configuration file, and `ifx_silent_deploy`, silently deploy a copy of the archived HCL OneDB™ server instance and its dbspaces on a target computer.

---

#### Related information

[Planning for deploying HCL OneDB on page 8](#)

## Introducing the HCL OneDB™ deployment assistant

The deployment assistant is used to create snapshots of HCL OneDB™ instances and their associated dbspaces, which can be later deployed on multiple target computers using the deployment utility.

The deployment assistant also allows users to reduce the footprint of the HCL OneDB™ installation by displaying file associations at the time of packaging.

### Using the deployment assistant

The deployment assistant is included on the HCL OneDB™ server at `$ONEDB_HOME/bin` on Linux™ and at `%ONEDB_HOME%\bin` on Windows™. To use the deployment assistant, you must run the following command from a command line:

```
ifxdeployassist
```

---

#### Related information

[Creating a snapshot with the deployment assistant on page 9](#)

## Introducing the HCL OneDB™ deployment utility

You can use the deployment utility to deploy snapshots of pre-configured instances and related dbspaces onto target computers.

You must install HCL OneDB™ before creating a snapshot. You can choose to include a fully configured server instance in the snapshot or create the instance at the time of deployment. The advantage of using the deployment utility is that you do not need to install and configure HCL OneDB™ on each target computer. You simply deploy a previously installed and fully configured instance. This gives you the ability to quickly recreate a fully configured instance on one or more target computers.

## Using the deployment utility

The `ifxdeploy` utility is included with the HCL OneDB™ server at `$ONEDB_HOME/bin` on Linux™ and UNIX™ and at `%ONEDB_HOME%\bin` on Windows™. To use the deployment utility, you must run the `ifxdeploy` utility either from a command line or as part of an application integration script. You can run the utility in silent mode without any user interaction.

### The `ifxdeploy` utility

The `ifxdeploy` utility is located in `$ONEDB_HOME/bin` on Linux™ and UNIX™ and at `%ONEDB_HOME%\bin` on Windows™.

To use the deployment utility, you must run the `ifxdeploy` utility either from a command line or as part of an application integration script. You can run the utility in silent mode without any user interaction.

### The deployment utility configuration file

You can pass parameters to the deployment utility using command-line options, or by using a configuration file that you pass as a parameter with the `-config` command-line option. You can also use both command-line options and a configuration file. Some examples on using `ifxdeploy` with command-line options and a configuration file on Linux™ and Windows™ are given below:

#### Example 1: Invoking the deployment utility with command-line options and a configuration file on Linux™ and UNIX™

```
ifxdeploy -y -verbose -config ifxdeploy.conf -file
/opt/IBM/Informix/demo_on.tgz -start 180
```

#### Example 2: Invoking the deployment utility with command-line options and a configuration file on Windows™

```
ifxdeploy.exe -y -verbose -config ifxdeploy.conf -file
C:\informixtemp\demo_on.zip -start 180
```

The following points summarize key information about the deployment utility configuration file:

- A sample deployment utility configuration file named `ifxdeploy.conf` is included on the HCL OneDB™ server at `$ONEDB_HOME/etc` on Linux™ and UNIX™ and at `%ONEDB_HOME%\etc` on Windows™.
- You can also specify instance-specific information for the instance that is getting deployed in the configuration file:
  - ONEDB\_SERVER sets the primary server name.
  - SERVERNUM sets the primary server port number.
  - ONEDB\_SQLHOSTS sets the value for the ONEDB\_SQLHOSTS environment variable.
  - ONCONFIG sets the `onconfig` file. If not specified and the **ONCONFIG** environment variable is not set, a new `onconfig` file is created based on the `onconfig.std` file template.
- Use `BEGIN ALIAS` and `END ALIAS` statements to define additional server names and listeners. Each alias results in a new `SQLHOSTS` entry and a new value for the `DBSERVERALIASES` configuration parameter in the `onconfig` file.
- Use the `BEGIN ONCONFIG` and `END ONCONFIG` statements to add to or override configuration parameters values specified in the `onconfig` file.

#### Example 3: Sample deployment utility configuration file entries for Windows™

```
ONEDB_SERVER demo_on
PROTOCOL1 onsoctcp
PORT1 9088
```

```

SERVERNUM 1

BEGIN ALIAS
SERVERNAME alias1
PROTOCOL drsoctcp
PORT 9091
END ALIAS

ONEDB_HOME "C:\Program Files\IBM\Informix"

RELOCATE C:\IFMXDATA\demo_on=C:\IFMXDATANEW\demo_on

ROOTPATH C:\IFMXDATANEW\demo_on\rootdbs_dat.000

BEGIN ONCONFIG
BUFFERPOOL default,buffers=1000,lrus=8,
lru_min_dirty=50.000000,lru_max_dirty=60.500000
MAX_PDQPRIORITY 80
END ONCONFIG

```

## Dynamic chunk relocation

The deployment utility provides the option to relocate the dbspaces to a new location of your choice. This option is particularly useful when the drives or the directory structures of the source template computer and the target computer are different. An example of this would be if the data spaces are on `/data1` on the template computer and you want to relocate them to `/data2` on the target computer.

You can set the new location of the dbspaces either by using the `-relocate` command-line parameter with `ifxdeploy`, or by using the configuration parameter `RELOCATE` in the `ifxdeploy.conf` file. You can specify a single new parent directory or map multiple separate chunk paths to different locations.

### Example 4: Command-line argument to relocate all chunk paths to a single directory

```
/data2/demo_on
```

```
-relocate /data2/demo_on
```

You can set the new location of the dbspaces either by using the `-rootpath` command-line parameter with `ifxdeploy`, or by using the configuration parameter `ROOTPATH` in the `ifxdeploy.conf` file. If you relocate dbspaces, you must also relocate the root dbspace.

### Example 5: Command-line argument to specify a new root path

```
-rootpath /data2/demo_on/online_root
```

---

## Related information

[Deploying HCL OneDB on page 7](#)

## Automating deployments of reduced-footprint HCL OneDB™ on Linux™

This section contains information about installing HCL OneDB™, creating an instance, using the deployment assistant to create a snapshot of the installation, and silently deploying the snapshots on Linux™.

### Perform HCL OneDB™ installation and create an instance on Linux™

The first task in the embeddability tutorial is to complete an installation and create the HCL OneDB™ instance.

Install HCL OneDB™ on the template computer to prepare the master copy for later deployment. A custom installation is recommended because you can choose the features to install. This is one good way to reduce the footprint of HCL OneDB™. Some features are mutually dependent and must be installed with one another. The good news is that the installation application manages these interdependencies.

 **Tip:** You will be able to further reduce the footprint of the installation when you create a snapshot of the HCL OneDB™ instance with the deployment assistant by clearing features and packages that you do not want to include in the snapshot.

The embeddability tutorial for Linux™ is based on the following installation scenario:

- HCL OneDB™ 1.0.0.0 is installed in the default directory `/opt/IBM/informix`
- An instance with the name `demo_on` is created

 **Important:** The `/opt/IBM/informix` installation path and an instance named `demo_on` are assumptions made for the tutorial. You can install HCL OneDB™ in a directory of your choice and create an instance with the name of your choice.

### Creating a snapshot for deployment on Linux™

The second task in the embeddability tutorial is to create a snapshot of the HCL OneDB™ install binary files and the dbspaces on the template computer by using the deployment assistant.

#### Before you begin

*Prerequisites:*

- The HCL OneDB™ installation and an instance of it are on a template computer.
- You must be logged in to the template computer as user **informix**.
- Save copies of `$ONEDB_HOME/bin/ifixdeploy` and `$ONEDB_HOME/etc/ifixdeploy.conf` outside of the `$ONEDB_HOME` directory. (You will need to have these copies outside of `$ONEDB_HOME` when you deploy the snapshot on the target computer.)
- The HCL OneDB™ instance from which you want to create the snapshot is running.

1. Start a terminal window.
2. Set the **ONEDB\_HOME** and **ONEDB\_SERVER** environment variables.

**Example**

For example:

```
export ONEDB_HOME=/opt/IBM/informix
export ONEDB_SERVER=demo_on
```

3. Start the deployment assistant with the following commands:

```
cd $ONEDB_HOME/bin
ifxdeployassist
```

4. Enter database server connection parameters in the first deployment assistant GUI window.

If you are completing this task as part of the embeddability tutorial, specify the following information:

- a. Instance Name: `demo_on`
- b. Host Name: `localhost`
- c. Port Number: `9088`
- d. Connection Type: `SQLI`

5. Specify the full path name of the snapshot file.

Enter `/home/informix/Desktop/demo_on` if you are completing this task as part of the embeddability tutorial.

6. **Optional:** Clear the features or packages that you do not want to include in the snapshot to reduce the footprint.



**Important:** The deployment assistant does not enforce any interdependencies between components of a functional instance.

7. If you are prompted with the Data Spaces window, decide whether to include the dbspaces associated with the source HCL OneDB™ instance in the snapshot. Keep the **Include the Data Spaces** box checked if you are doing this task as part of the embeddability tutorial.



**Tip:** Record the full path names of the dbspaces if you select to include them in the snapshot. You will need to know the locations of the dbspaces when you deploy the snapshot.

8. Review the deployment configuration summary.
9. Record the information displayed in the Deployment Summary window and click **Close**. Save the information that you have recorded because you will need to know the snapshot locations for the next step.
10. Save the snapshots that you have created with the deployment assistant, the `$ONEDB_HOME/bin/ifxdeploy` file, and the `$ONEDB_HOME/etc/ifxdeploy.conf` file on external media (for example, a flash drive or CD) for future deployment.

For the embeddability tutorial: save the database server snapshot as `demo_on.tar.gz` and the dbspaces snapshot as `demo_on_db.tar.gz`.

---

**Related information**

[Easy deployment on page 4](#)

## Silently deploy snapshots on Linux™

The third task of the embeddability tutorial is to silently deploy the HCL OneDB™ snapshot.

### Before you begin

Deploying the HCL OneDB™ instance and its dbspaces is a two-step process:

1. Deploy dbspaces from the snapshot that was created using the deployment assistant.
2. Deploy the server instance, optionally relocate the dbspaces, and start the instance by using the deployment utility.

You can automate this process by using a simple shell script. The `ifx_silent_deploy.sh` script is an example shell script which serves this purpose.

Prerequisites:

- You must be logged in to the target computer as root user.
- You must have a snapshot of the HCL OneDB™ instance and any associated dbspaces that was created with the deployment assistant.
- The **ONEDB\_HOME** environment variable must be set.
- A folder named `informixtemp` must exist on the target computer.
- All the dbspaces' chunks must be cooked files, and they must be located in a single directory.
- The following files and utilities must be in the `informixtemp` folder on the target computer:
  - HCL OneDB™ instance snapshot (for example: **demo\_on.tgz**)
  - dbspaces snapshot (for example: **demo\_on\_db.tgz**)
  - deployment utility (`ifxdeploy`)
  - deployment utility configuration file (`ifxdeploy.conf`)
  - the `ifx_silent_deploy.sh` shell script to complete silent deployment

1. Run the `chmod +x ifx_silent_deploy.sh` command to give execute permission to the root user to run the `ifx_silent_deploy.sh` script.
2. Configure the following `ifxdeploy.conf` file parameters.
  - a. Set the `ONEDB_SERVER`, `ONCONFIG`, and `ONEDB_SQLHOSTS` parameters. Alternatively, these parameters can be set as environment variables.
  - b. Set the user **informix** password on the target computer in the `INFORMIXPASSWORD` parameter of the `ifxdeploy.conf` file.
  - c. **Optional:** Set other parameters in the `ifxdeploy.conf` file as needed for your environment.
3. Use the following information about the `ifx_silent_deploy.sh` script to silently deploy the HCL OneDB™ instance. After the syntax information, there are two examples given that indicate specific values if you are completing the embeddability tutorial.

*Syntax:*

```
ifx_silent_deploy.sh <relocate_option> <srvpkg> <dbspkg> <currloc> [<newloc>]
```

- `relocate_option`: use `relocate` to relocate dbspaces and `norelocate` to not relocate dbspaces
- `srvpkg`: name of the server package
- `dbspkg`: name of the dbspaces package
- `currloc`: current location of the dbspaces
- `newloc`: new deployment location of the dbspaces when relocating. This argument is not required if you are not relocating dbspaces.

### Example



**Important:** The following commands are only examples, such as if you are using this documentation with sample values to complete the embeddability tutorial. The names of the snapshot files and location of dbspaces might be different in your instance.

### Choose from:

- To silently deploy the HCL OneDB™ instance and relocate the dbspaces:

```
ifx_silent_deploy.sh relocate demo_on.tgz demo_on_db.tgz
/data/IBM/informix/demo/server /opt/IBM/data/demo_on
```

This command deploys the base server of the snapshot, relocates the dbspaces from `/data/IBM/informix/demo/server` to `/opt/IBM/data/demo_on`, and starts the deployed instance.

- To silently deploy the HCL OneDB™ instance without relocating the dbspaces:

```
ifx_silent_deploy.sh norelocate demo_on.tgz demo_on_db.tgz
/data/IBM/informix/demo/server
```

This command deploys the HCL OneDB™ base server instance, deploys associated dbspaces to `/data/IBM/informix/demo/server`, and starts the instance.

### Results

See the `informixtemp/ifxdeploy.log` file for messages that are logged by the deployment utility. If the HCL OneDB™ instance does not start automatically, the probable cause is that during creation of the snapshot you removed a feature or package that is required by the base server instance to run.

The deployed instance does not require the `ifxdeploy.conf` file to function. You can delete the `ifxdeploy.conf` file from the `informixtemp` directory after deploying the HCL OneDB™ instance.

## Automating deployments of reduced-footprint HCL OneDB™ on Windows™

This section contains information about installing HCL OneDB™, creating an instance, using the deployment assistant to create a snapshot of the installation, and silently deploying the snapshots on Windows™.

### Perform HCL OneDB™ installation and create an instance on Windows™

The first task in the embeddability tutorial is to complete an installation and create the HCL OneDB™ instance.

Install HCL OneDB™ on the template computer to prepare the master copy for later deployment. A custom installation is recommended because you can choose which features to install. This is one good way to reduce the footprint of HCL OneDB™. Some features are mutually dependent and must be installed with one another. The good news is that the installation application manages these interdependencies.

 **Tip:** You will be able to further reduce the footprint of the installation when you create a snapshot of the HCL OneDB™ instance with the deployment assistant by clearing features and packages that you do not want to include in the snapshot.

The embeddability tutorial for Windows™ is based on the following installation scenario:

- HCL OneDB™ 1.0.0.0 is installed in the default directory `C:\Program Files\IBM\Informix`
- An instance with the name `demo_on` is created

 **Important:** The `C:\Program Files\IBM\Informix` installation path and an instance named `demo_on` are assumptions made for the tutorial. You can install HCL OneDB™ in a directory of your choice and create an instance with the name of your choice.

## Creating a snapshot for deployment on Windows™

The second task in the embeddability tutorial is to create a snapshot of the HCL OneDB™ install binary files and the dbspaces on the template computer by using the deployment assistant.

### About this task

Prerequisites:

- The HCL OneDB™ installation and an instance of it are on a template computer.
- You must be logged in to the template computer as user **informix** or as a user with Administrator privileges.
- In order to package dbspaces, the user must have CONNECT privileges to the **sysadmin** database. By default, only the user **informix** has this permission. If you log in as a user other than user **informix**, you must grant the CONNECT privilege to the **sysdamin** database for this user. This is a security mechanism implemented to protect the data.
- Save copies of the files `%ONEDB_HOME%\bin\ifxdeploy.exe` and `%ONEDB_HOME%\etc\ifxdeploy.conf` files in a folder outside `%ONEDB_HOME%`. You will need these files when you deploy HCL OneDB™ on the target computer.
- The HCL OneDB™ instance from which you want to create the snapshot is running.

1. Start a command window.
2. Set the **ONEDB\_HOME** and **ONEDB\_SERVER** environment variables.

### Example

For example, for the embeddability tutorial, specify:

```
set ONEDB_HOME=C:\Program Files\IBM\Informix
set ONEDB_SERVER=demo_on
```

3. Start the deployment assistant with the following command.

For example, for the embeddability tutorial, specify:

```
cd C:\Program Files\IBM\Informix\bin
ifxdeployassist
```

4. Verify that the deployment assistant detects the HCL OneDB™ instance to be used as a template for the snapshot.
5. Specify the full path name of the snapshot to be created.

For the embeddability tutorial, specify `C:\Documents and Settings\Administrator\Desktop\demo_on.zip`.

6. **Optional:** Clear the features or packages that you do not want to include in the snapshot to reduce the footprint.



**Important:** The deployment assistant does not enforce any interdependencies between components of a functional instance.

7. If you are prompted with the Data Spaces window, decide whether to include the dbspaces associated with the template HCL OneDB™ instance in the snapshot. Keep the **Include the Data Spaces** box checked if you are creating a snapshot following the embeddability tutorial.



**Tip:** Record the full path names of the dbspaces if you select to include them in the snapshot. You will need to know the locations of the dbspaces when you deploy the snapshot.

8. Review the deployment configuration summary.
9. Record the information displayed in the Deployment Summary window and click **Close** . Save the information that you have recorded because you will need to know the snapshot locations for the next step.
10. Save the snapshots that you created with the deployment assistant, the `%ONEDB_HOME%\bin\ifxdeploy.exe` file and the `%ONEDB_HOME%\etc\ifxdeploy.conf` file on external media (for example, a flash drive or CD). You will need these for the next task.

For the embeddability tutorial: save the database server snapshot as **demo\_on.zip** and the dbspaces snapshot as **demo\_on\_db.zip**.

## Silently deploy HCL OneDB™ on Windows™

The third task of the embeddability tutorial is to silently deploy the HCL OneDB™ snapshot.

### Before you begin

Deploying the HCL OneDB™ instance and its dbspaces is a two-step process:

1. Deploy dbspaces from the snapshot that was created using the deployment assistant.
2. Deploy the server instance, optionally relocate the dbspaces, and start the instance by using the deployment utility.

You can automate this process by using a simple batch script. The `ifx_silent_deploy.cmd` script is an example batch script which serves this purpose.

## Prerequisites:

- You must be logged in to the target computer as an Administrator user.
- You must have a snapshot of the HCL OneDB™ instance and any associated dbspaces that was created with the deployment assistant.
- Installation of 7-zip on the target computer. If you need to install 7-zip, download and install it on the target computer at the default location `C:\Program Files\7-Zip`.
- A folder named `informixtemp` must exist on the `C:\` drive of the target computer.
- The **ONEDB\_HOME** environment variable must be set.
- All the dbspaces' chunks must be cooked files, and they must be located in a single directory.
- The following files and utilities must be in the `C:\informixtemp` folder on the target computer:
  - HCL OneDB™ server instance snapshot (Example: `demo_on.zip`)
  - dbspaces snapshot (Example: `demo_on_db.zip`)
  - deployment utility (`ifxdeploy.exe`)
  - deployment utility configuration file (`ifxdeploy.conf`)
  - the `ifx_silent_deploy.cmd` script to complete silent deployment

1. Configure the following `ifxdeploy.conf` file parameters.
  - a. Set the `ONEDB_SERVER` and `ONCONFIG` parameters. Alternatively, these parameters can be set as environment variables.
  - b. Set the user **informix** password on the target computer in the `INFORMIXPASSWORD` parameter of the `ifxdeploy.conf` file.
  - c. **Optional:** Set other parameters in the `ifxdeploy.conf` file as needed for your environment.
2. Use the following information about the `ifx_silent_deploy.cmd` script to silently deploy the HCL OneDB™ instance. After the syntax information, there are two examples given that indicate specific values if you are completing the embeddability tutorial.

## Syntax:

```
ifx_silent_deploy.cmd <relocate_option> <srvpkg> <dbspkg> <currloc> [<newloc>]
```

- `relocate_option`: use `relocate` to relocate dbspaces and `norelocate` to not relocate dbspaces
- `srvpkg`: name of the server package (for example: **demo\_on.zip**)
- `dbspkg`: name of the dbspaces package (for example: **demo\_on\_db.zip**)
- `currloc`: current location of the dbspaces
- `newloc`: new deployment location of the dbspaces when relocating. This argument is not required if you are not relocating dbspaces.

## Example



**Important:** The following commands are only examples, such as if you are using this documentation with sample values to complete the embeddability tutorial. The names of the snapshot files and location of dbspaces might be different in your instance.

## Choose from:

- To silently deploy the HCL OneDB™ instance and relocate the dbspaces:

```
ifx_silent_deploy.cmd relocate demo_on.zip demo_on_db.zip
C:\IFMXDATA\demo_on C:\IFMXDATANEW\demo_on
```

This command deploys the HCL OneDB™ instance, relocates dbspaces from C:\IFMXDATA\demo\_on to C:\IFMXDATANEW\demo\_on and starts the instance.

- To silently deploy the HCL OneDB™ instance without relocating the dbspaces:

```
ifx_silent_deploy.cmd norelocate demo_on.zip demo_on_db.zip
C:\IFMXDATA\demo_on
```

This command deploys the HCL OneDB™ instance, deploys dbspaces to C:\IFMXDATA\demo\_on, and starts the instance.

## Results

See the C:\informixtemp\ifxdeploy.log file for messages that are logged by the deployment utility. If the HCL OneDB™ instance does not start automatically, the probable cause is that during creation of the snapshot you removed a feature or package that is required by the base server instance to run.

You can delete the ifxdeploy.conf file from C:\informixtemp after deploying the HCL OneDB™ instance.

## Manage memory in embedded environments

You can manage memory in embedded environments by enabling the database server to automatically rotate and delete online message log files, configuring the actions that the server takes to continue processing when memory is critically low, and enabling the server to reserve a specific amount of memory for use when critical rollback activities are needed and the server has limited free memory.

## Optimize storage

You can optimize storage by compressing data, configuring the server to add more storage space automatically, and defragmenting partitions. Compressing data reduces the amount of disk space needed for your applications.

Use SQL administration API commands to:

- Compress data in table or fragment row and simple large objects in dbspaces, consolidate the remaining free space, and return the free space to the dbspace.
- Compress only row data or only simple large objects in dbspaces.
- Compress detached B-tree indexes, consolidate free space in the index, and return the free space to the dbspace. You can also compress indexes when you use create them with the CREATE INDEX statement.
- Implement automatic space management
- Defragment partitions to non-contiguous extents

- Compress data in table or fragment rows
- Consolidate the free space that remains in the table or fragment
- Return the free space to the dbspace.
- Implement automatic space management
- Defragment partitions to non-contiguous extents

---

#### Related information

[Compression on page](#)

[B-tree index compression on page](#)

[Automatic space management on page](#)

[Partition defragmentation on page](#)

## Manage message logs in an embedded environment

You can enable the database server to automatically rotate and delete online message log files, and you can specify the maximum number of online, ON-Bar activity, or ON-Bar debug message log files to rotate. The Scheduler tasks that rotate and delete these message log files are useful for embedded applications, because they reduce DBA or system administrator requirements for managing the logs.

Additionally, you can run SQL administration API commands to rotate, remove the contents of (*truncate*), or delete one of these message log files. For example, you might want to run a truncate or delete command when performing a load operation.

## Enabling and disabling the tasks that automatically rotate message log files

Before the database server can run the built-in Scheduler tasks that automatically rotate message log files, you must enable each task that rotates a message log file. You only need to enable each task once. You can also disable an enabled task.

### About this task

To enable a task for rotating logs:

Run the following SQL statement, which updates the value of the **tk\_enable** column of the **ph\_task** table to **T** (true) for the message log file where the **tk\_name** column is the name of the message log, as follows:

```
UPDATE ph_task SET tk_enable = t , tk_frequency = frequency
WHERE tk_name = name_of_message_log task";
```

The name of the message log must be `bar_act_log_rotate`, `bar_debug_log_rotate`, or `online_log_rotate`.

For example, to enable the **online\_log\_rotate** task and set the frequency of the task to every 30 days, specify:

```
DATABASE sysadmin;
UPDATE ph_task SET tk_enable = "t" ,
tk_frequency = INTERVAL (30) DAY TO DAY
WHERE tk_name = "online_log_rotate";
```

You can also specify `tk_frequency = tk_frequency` if you do not want to accept the default frequency (3:00 A.M. every 30 days) when you enable the task.

**What to do next**

The only possible values for `tk_enable` are `t` and `f` (true and false).

To disable the running of a message log rotation task, set `tk_enable` to `f`, as shown in the following example:

```
UPDATE ph_task SET tk_enable = "f" ,
tk_frequency = INTERVAL (30) DAY TO DAY
WHERE tk_name = "bar_debug_log_rotate";
```

You can also specify `tk_frequency = tk_frequency` if you want to accept the default frequency (3:00 A.M. every 30 days).

**Tasks that automatically rotate message log files**

A Scheduler task automatically rotates each of the common message log files (`online.log`, `bar_act_log`, and `bar_debug_log`). When the server rotates a log file, the server switches to a new log file and increments the ID numbers for the previous log files by one. When the maximum number of log files is reached, the log file with the highest ID is deleted.

The following table shows the tasks that automatically rotate message log files and the default frequency for when the tasks run:

**Table 4. Message log files, tasks that rotate the files, and default frequency**

Message log file	Task that rotates the file	Default frequency for when task runs
<code>online.log</code>	<b>online_log_rotate</b> task	3:00 A.M. every 30 days  You can change the frequency, which is specified in the <code>MAX_MSGPATH_VERSIONS</code> parameter of the <b>ph_threshold</b> table.
<code>bar_act_log</code>	<b>bar_act_log_rotate</b> task	3:00 A.M. every 30 days  You can change frequency, which is specified in the <code>MAX_BAR_ACT_LOG_VERSIONS</code> parameter of the <b>ph_threshold</b> table.
<code>bar_debug_log</code>	<b>bar_debug_log_rotate</b> task	3:00 A.M. every 30 days  You can change frequency, which is specified in the <code>MAX_BAR_DEBUG_LOG_VERSIONS</code> parameter of the <b>ph_threshold</b> table.

You can change the maximum number of online, ON-Bar activity, or ON-Bar debug message log files that the database server retains. This information is stored in the `MAX_MSGPATH_VERSIONS`, `MAX_BAR_ACT_LOG_VERSIONS`, and `MAX_BAR_DEBUG_LOG_VERSIONS` parameters of the `ph_threshold` table.

Before the server can run the tasks that automatically rotate message log files, you must enable each task. You only need to enable each task once.

## Configuring the frequency of automatic rotation of message log files

You can configure the frequency of the built-in Scheduler tasks that automatically rotate online message log files.

### About this task

The default frequency of the **bar\_act\_log\_rotate**, **bar\_debug\_log\_rotate**, and **online\_log\_rotate** tasks is 3 A.M. every 30 days with a maximum of 12 log files, but you can configure the task to run more or less frequently.

### Prerequisite:

If you did not enable the task for rotating the message log that you want to configure, do that now.

### To configure a task for rotating logs to run more or less frequently:

Run the following SQL statements, where *days* is the number of days between each run:

```
DATABASE sysadmin;
UPDATE ph_task set tk_frequency = frequency
WHERE tk_name = "task_name";
```

The task name must be `bar_act_log_rotate`, `bar_debug_log_rotate`, or `online_log_rotate`.

For example, to change the frequency of the **online\_log\_rotate** task specify:

```
DATABASE sysadmin;
UPDATE ph_task SET tk_frequency = INTERVAL (10)
DAY TO DAY WHERE tk_name = "online_log_rotate";
```

### What to do next

You can also disable the running of message log rotation tasks, by setting `tk_enable` to `f` (false), as follows:

```
UPDATE ph_task SET tk_enable = f , tk_frequency = tk_frequency
WHERE tk_name = "online_log_rotate";
```

## Configuring the maximum number of message log files to retain

You can change the maximum number of online, ON-Bar activity, or ON-Bar debug message log files that the database server retains. This information is stored in the `MAX_MSGPATH_VERSIONS`, `MAX_BAR_ACT_LOG_VERSIONS`, and `MAX_BAR_DEBUG_LOG_VERSIONS` parameters of the **ph\_threshold** table.

### About this task

By default, the database server retains 12 online, ON-Bar activity, and ON-Bar debug message log files.

### To change the maximum number of online log files to retain:

Run the following SQL statements to specify the maximum value of log files to retain, where *threshold\_name* is `MAX_MSGPATH_VERSIONS` (for the online log files), `MAX_BAR_ACT_LOG_VERSIONS`, or `MAX_BAR_DEBUG_LOG_VERSIONS`.

```
UPDATE ph_threshold
SET value = "number"
WHERE name = "threshold_name";
```

For example, to set the max number of online log files to keep to 20, specify:

```
UPDATE ph_threshold
SET value = "20"
WHERE name = "MAX_MSGPATH_VERSIONS";
```

## Reducing the size of message log files

You can reduce the size of online, ON-Bar activity, or ON-Bar debug message log files by running SQL administration API commands that rotate, truncate, or delete the logs.

### About this task

To manually rotate a message log file:

1. Run the `admin()` or `task()` function with the **message log rotate** argument, the name of the message log, and the maximum message log version that the server will rotate, as follows:

```
EXECUTE FUNCTION task("message log rotate",
  "full_path_for_message_log_file", maximum_version);
```

For example, to rotate a maximum of 52 `online.log` files, specify:

```
execute function task("message log rotate",
  "/usr/informix/online.log",52);
```

To manually truncate a message log file (remove the contents of the file):

1. Run the `admin()` or `task()` function with the **message log truncate** argument and the full path name for the message log file, as follows:

```
EXECUTE FUNCTION task("message log truncate",
  "full_path_for_message_log_file");
```

For example, to remove the contents of the `online.log` file, specify:

```
EXECUTE FUNCTION task("message log truncate",
  "/usr/informix/online.log");
```

To manually delete a message log file (remove the contents of the file and the file):

1. Run the `admin()` or `task()` function with the **message log delete** argument and the full path name for the message log file, as follows:

```
EXECUTE FUNCTION task("message log delete",
  "path_for_message_log_file");
```

For example, to delete the entire `online.log` file, specify:

```
EXECUTE FUNCTION task("message log delete",
"/usr/informix/online.log");
```

---

#### Related information

[message log rotate argument: Rotate the message log file \(SQL administration API\) on page](#)

[message log truncate argument: Delete the contents of a message log file \(SQL administration API\) on page](#)

[message log delete argument: Delete a message log file \(SQL administration API\) on page](#)

## Reserve memory for critical activities in embedded environments

You can reserve a specific amount of memory for use when critical activities (such as rollback activities) are needed and the database server has limited free memory. When you reserve memory, critical activities, such as rollback activities, can complete even when you receive out-of-memory errors.

To reserve memory for critical activities, set the `LOW_MEMORY_RESERVE` configuration parameter to a specified value in kilobytes. You can also change the `LOW_MEMORY_RESERVE` value by using the **onmode -wf** or **onmode -wm** utility.

---

#### Related information

[LOW\\_MEMORY\\_RESERVE configuration parameter on page](#)

[Reserve memory for critical activities on page](#)

## Maintain a targeted amount of memory in embedded applications

You can configure the database server to continue processing when memory is critically low, instead of returning an out of memory error.

You specify the criteria for terminating sessions based on idle time, memory usage, and other factors so that the targeted application can continue to process. Configuring the low memory response is useful for embedded applications that have memory limitations.

To configure a targeted amount of memory in embedded applications:

- Set the `LOW_MEMORY_MGR` configuration parameter to `1`, which enables the automatic low memory management when the server starts.
- Set the threshold parameters for the amount of memory to maintain by using an SQL administration API command with the **scheduler lmm enable** argument.

To disable automatic low memory management, run an SQL administration API command with the **scheduler lmm disable** argument.

**Related information**

[LOW\\_MEMORY\\_MGR configuration parameter on page](#)

[scheduler Imm enable argument: Specify automatic low memory management settings \(SQL administration API\) on page](#)

[onstat -g Imm command: Print low memory management information on page](#)

[Configure the server response when memory is critically low on page](#)

# Index

## A

- Administration
  - automating 5

## C

- changing the frequency 35
- client connection
  - synchronizing with server 4
- clone 7
- Command line
  - deploying
    - HCL OneDB
  - snapshot 15
- Configuration 5
- Configuration parameters
  - LOW\_MEMORY\_MGR configuration parameter 37
  - LOW\_MEMORY\_RESERVE configuration parameter 37

## D

- Deploying
  - an instance 4
- Deploying HCL OneDB 8
- deploying to multiple computers 7
- Deployment
  - planning 8
- deployment assistant 22
  - command 11
  - creating a snapshot 9
- Deployment assistant 21
- deployment utility 7, 22
  - removing a snapshot 21
  - syntax and options 15
- Deployment utility 8
- disabling rotation tasks 33

## E

- Embeddability Toolkit
  - components 21
  - create a snapshot on Linux 25
  - create a snapshot on Windows 29
  - deploy snapshots on Linux 27
  - deploy snapshots on Windows 30
  - HCL OneDB
  - snapshots
    - deploying on Linux 27
  - install on Linux 25
  - install on Windows 28
  - tutorial 21, 25, 25, 27, 28, 29, 30
- Embedded applications
  - maintaining targeted memory 37
  - reserving memory 37
- enabling rotation tasks 33
- Environment 6

## H

- HCL OneDB
- snapshots
  - deploying on Windows 30
- High-availability overview 7

## I

- ifxdeploy command 14
  - examples on UNIX 15
  - examples on Windows 15

- ifxdeploy utility 22
- ifxdeployassist command 11, 22
- Installation
  - customizing 3

## L

- LOW\_MEMORY\_MGR configuration parameter 37
- LOW\_MEMORY\_RESERVE configuration parameter 37

## M

- managing 5, 32, 33, 34
- Memory 32
  - reserving 37
  - targeted amount 37
- Message logs 33, 34
  - changing the threshold 35
  - managing 33, 35, 35
  - rotation 33, 35, 35

## O

- onconfig file
  - specifying environment 6

## R

- Response file 3

## S

- server connection
  - synchronizing with client 4
- Silent installation
  - customizing 3
- snapshot 7
  - creating for deployment 9, 13
  - deploying 7, 14
  - removing 21
- specifying in onconfig file 6
- sqlhosts file 4
- sync\_registry task
  - about 4
- syncsqlhosts utility
  - about 4
  - options 19
  - syntax 19
  - usage 19

## U

- Utilities
  - syncsqlhosts 19
  - syncsqlhosts syntax 19