Part XIII: Embedding Informix

IBM

Tables of Contents

Embedding Informix database servers

When you embed IBM® Informix®, you can use enterprise-class high-availability and high performance with embeddability features such as easy programmability, a small disk and memory footprint, and silent deployment.

External resources

- Smart Systems Require an Embeddable Database: Six Criteria to Consider (White paper)
  This Cisco and Strategic Alliance paper discusses the installation, integration, performance, resource management, administration, and availability criteria of embedded databases.
- Embedded Informix: Think Inside the Box (IBM developerWorks®)
  A discussion of the reliability and performance characteristics of that make it an excellent candidate for use as an embedded database server, and a walkthrough of creating an embedded deployment. provides features, such as the deployment wizard and deployment utility, that simplify deploying a database server in an embedded environment.

- Embedding Informix
  These topics describe how to embed Informix in your applications.

Embedding IBM Informix

These topics describe how to embed IBM® Informix® in your applications.

This information is intended for application programmers.

- Overview of embedding Informix products
  You can embed 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.
- Deploying Informix software
  The product supports embeddability tools that enable lightweight, faster deployment of the database server and related client programs.
- Informix embeddability deployment tutorial
  The purpose of this tutorial is to provide steps and scripts needed for silent end-to-end deployment of IBM Informix on Linux and Windows using the deployment assistant and the deployment utility.
- 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.

Overview of embedding Informix products

You can embed 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 Informix
  If your application runs in a low-memory environment, you might need to minimize the size of IBM Informix.
- Customize a silent installation
  You can create a silent installation script, customized for your requirements, that you can use multiple times.

- Easy deployment
  You can use the deployment tools to quickly and easily deploy IBM Informix to multiple locations.
- Synchronize connectivity information between servers and clients on Windows
  On the Windows operating system, you must configure connectivity information for Informix Servers by using the sqlhosts file.
- 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.
- 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.
- Specify the Informix environment in the onconfig file
  You can set local environment variables in the onconfig file for the Informix 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**, **onload** and **onunload**, **onlog**, **onmode**, **onparams**, **onspaces**, **onstat**, and **ontape** utilities in embedded environments.
- Maintain Informix availability
  You can ensure that your applications can always access IBM Informix even if a single server fails by configuring a high-availability solution.

Customize the size of Informix

If your application runs in a low-memory environment, you might need to minimize the size of IBM® Informix®.

You can customize the size of IBM Informix during installation and during snapshot creation.

You can use the Deployment wizard during installation to exclude parts of the 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 installation is approximately 100 MB.

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

Related concepts:
The deployment utility
Related information:
Exclude database server components

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 IBM® Informix® 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 *IBM Informix Installation Guide*.

Related information:
Installation owner
Running a silent installation

Easy deployment

You can use the deployment tools to quickly and easily deploy IBM® Informix® to multiple locations.

You can take a snapshot of a fully configured 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 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 tasks:
Creating a snapshot with the deployment assistant
Creating a snapshot for deployment on Linux

Synchronize connectivity information between servers and clients on Windows

On the Windows operating system, you must configure connectivity information for Informix® Servers by using the sqlhosts file.

The sqlhosts file is installed in %INFORMIXDIR%\etc\sqlhosts.%INFORMIXSERVER%. 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

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 IBM® Informix®:

- 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 :

- 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
The Scheduler
Remote administration with the SQL administration API
The ph_alert Table
onconfig Portal: Configuration parameters by functional category
Database server configuration during installation

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 concepts:
Specify the Informix environment in the onconfig file
Related information:
Database server configuration
onmode -wf, -wm: Dynamically change certain configuration parameters
onmode -we: Export a file that contains current configuration parameters
onmode -wi: Import a configuration parameter file
import config argument: Import configuration parameter values (SQL administration API)
export config argument: Export configuration parameter values (SQL administration API)
AUTO_TUNE configuration parameter

Specify the Informix environment in the onconfig file

You can set local environment variables in the onconfig file for the Informix® 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**, **onload** and **onunload**, **onlog**, **onmode**, **onparams**, **onspaces**, **onstat**, and **ontape** 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 IBM® Informix 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 concepts:

Related information:

Maintain Informix availability

You can ensure that your applications can always access IBM® Informix® even if a single server fails by configuring a high-availability solution.

Sometimes Informix-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:

Deploying Informix software

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

Use the command-line deployment utility to deploy a snapshot of IBM® Informix® 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 installation and then distribute the image to other locations on supported Linux platforms.

- The deployment utility
  The IBM Informix deployment utility can deploy snapshots of pre-configured IBM Informix instances (with or without data) on one or more computers.
- The syncsqlhosts utility
  Use the **syncsqlhosts** utility to manually convert the connection information between the sqlhosts file format and the Windows registry format.
- 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.
- Deploying Informix and client products with RPM-based installation (Linux)
  You can create RedHat Package Manager (RPM) images of a customized IBM Informix product installation to deploy the products with a silent installation. You can use RPM images to deploy the database server and client products in embedded applications. After you install the database server with an RPM image, you must manually initialize the database server.

Related concepts:

The deployment utility

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

A snapshot is an image of the 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 with the silent installation option. When you deploye 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 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.

- Planning for deploying Informix
  Before you deploy Informix, you should decide what you want to include in your deployment and how to configure the target server.
- Creating a snapshot with the deployment assistant
  Use the deployment assistant to create a snapshot of a running Informix 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.
- Creating a snapshot for deployment manually
  Create a snapshot of IBM Informix that you can use with the deployment utility to place pre-configured instances on multiple computers.
- Deploying a snapshot with the deployment utility
  Deploy a snapshot of an instance by using the **ifxdeploy** utility on the target computer.
- The ifxdeploy command: The deployment utility
  Use the **ifxdeploy** command to deploy a snapshot or remove a snapshot that you already deployed.

Related concepts:
Customize the size of Informix

Planning for deploying Informix

Before you deploy Informix®, 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 Informix installation on the machine, you can deploy a new installation using an archive of the INFORMIXDIR directory from your source machine. You can deploy with the following command:

```
ifxdeploy  -INSTALL -i informixdir --snapshot archive_file
```

  The *archive_file* is uncompressed in the directory specified by the **-i** parameter or the INFORMIXDIR 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 INFORMIXDIR, INFORMIXSERVER, 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 concepts:
Informix embeddability deployment tutorial
Related tasks:
Creating a snapshot with the deployment assistant

Creating a snapshot with the deployment assistant

Use the deployment assistant to create a snapshot of a running Informix® 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 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.
- You are able to run the deployment assistant on the computer that hosts the Informix 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.

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 %INFORMIXDIR%\bin (Windows) or $INFORMIXDIR/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.

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.

- 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.

Related concepts:
Easy deployment
Planning for deploying Informix
Introducing the Informix deployment assistant

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--+--------------------+-------------------->
                    '- -a--archive_format-'

>--+---------------+--+-----------------------------+-------->
   '- -f--path_name-'  '- -i--connectivity_information-'

>--+-----+--+-------------+--+-----+--+-----+----------------><
   '- -c-'  '- -d--+-----+-'  '- -h-'  '- -v-'
                   '- -n-'
```

Command options

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

Option	Purpose	Key Considerations
-a archive_format	Specifies the archive format and compression type to save the snapshot. The following values can be used in the place of *archive_format*: - BZIP2 (tar archive with BZip2 compression) - GZIP (tar archive with GZIP compression) - TAR (tar archive with no compression) - ZIP (zip archive with compression)	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.

Option	Purpose	Key Considerations
`-f path_name`	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: *server_name_yymmdd-hhmm*.tar.gz The *server_name* is the name of the server. The *yymmdd-hhmm* is the time the snapshot is taken. The default file name of the data snapshot is: *server_name_yymmdd-hhmm*_db.tar.gz 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:`ifxdeployassist -a TAR -f C:\Informix®\server.zip` then the snapshot will be saved as a tar archive with no compression rather than as a zip archive with compression.
`-i connectivity _information`	Specifies the connectivity information for the template instance from which to create a snapshot. If you do not use the `-i` option, the deployment assistant detects the connectivity information from the current environment and uses the same connectivity settings. In the place of *connectivity_information*, indicate specific connectivity settings for the snapshot to use after it is deployed. Set values for the following variables, separated by colons without spaces: • name of the database server instance • name of the template computer for which the database server instance is configured • service port number • protocol for the connection	The only valid values for the protocol information are `SQLI` and `DRDA`. If no protocol is specified, the command sets the connection to SQLI. The connectivity information here is contained in the sqlhosts file. **Example:** `ifxdeployassist -i hrapps:oscar:40:DRDA`
`-c`	Directs the deployment assistant to run in a command-line interface.	The `-c` 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.
`-d`	Includes the data associated with the instance in the snapshot.	
`-n`	Captures a snapshot without the IBM® Informix database server instance installation.	You must run the `-d` option in the command in order to use the `-n` option.
`-h`	Displays the online help for the **ifxdeployassist** command.	
`-v`	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 Informix database server exists in C:\informix.
- The **ifxdeployassist** command is run in C:\informix\bin.
- The INFORMIXSERVER 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 IBM® Informix® that you can use with the deployment utility to place pre-configured instances on multiple computers.

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.

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

1. Shut down the instance in a consistent state with **onmode -kuy**.
2. Create a snapshot of the following items:
   - 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 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 %INFORMIXDIR%\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.

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:
   - 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:
   - **-i** *INFORMIXDIR*
   - **-s** *INFORMIXSERVER*
   - **--ixpass** *password*
4. Run the **ifxdeploy** command with the appropriate options.
5. **Windows:** Manually add the IBM® Informix® .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--INFORMIXDIR--------------------->

>-- --snapshot--snapshot_file--+-----------+--+-----------+--->
                               '- --verbose-'  '- --nonroot-'

>--+----------+--------------------------------------------><
   '- --silent-'
```

```
Create instance command

>>-ifxdeploy-- -INSTANCE create-- -i--INFORMIXDIR-------------->

>-- -s--INFORMIXSERVER--+-------------+---------------------->
                        '- -o--ONCONFIG-'

>-- --ixpass--password--+--------------------------+--------->
                        '- --sqlhosts--sqlhosts_file-'

   .----------------------------------------------------------------------.
   V                                                                      |
>----+----------------------------------------------------------------+-+-+-->
     '- --dbalias--name--=--alias--,--protocol--=--protocol--,--port--=--port-'

>--+----------------------------+---------------------------->
   '- --servernum--server_number-'

>--+--------------------------------------------+-------->
   '- --image_archive--archive_file--,--dest_directory-'

>--+--------------------+------------------------------------>
   '- --rootpath--rootpath-'

>--+----------------------------------------------------------+-->
   '- --relocate--+-path------------------------------------+-'
                  | .-;----------------.                    |
                  | V                  |                    |
                  +---old_path=new_path-+--------------------+
                  | .-;--------------------------------------. |
                  | V                                        | |
                  '---old_path,old_offset=new_path,new_offset-+-'

>--+-----------------+--+---------+--+---------+-------->
   '- --start--number-'  '- --verbose-'  '- --nonroot-'

>--+----------+------------------------------------------------><
   '- --silent-'


Uninstallation command

>>-ifxdeploy-- -UNINSTALL-- -i--INFORMIXDIR--+-----------+----->
                                             '- --verbose-'

>--+-----------+--+-----------+--+----------+--------------><
   '- --deluser-'  '- --nonroot-'  '- --silent-'
```

Command options

Table 1. The **ifxdeploy** command options

Element	Purpose	Key considerations
--deluser	Windows: Removes user **informix** and the **Informix-Admin** group. Linux and UNIX: Removes user **informix** and group **informix** or the non-root user who owns the instance.	The **--delifx** option functions when there is only one IBM® Informix® installation on the computer.

Element	Purpose	Key considerations
--dbalias name = *alias*, **protocol**=*protocol*, **port**=*port*	Specifies new database server aliases and related SQLHOSTS connectivity settings for the deployed instance.	Each *alias* name must be unique and conform to server alias name rules. See DBSERVERALIAS configuration parameter. The *protocol* must be a valid connection type for the operating system. The *port* must be a valid port number. See sqlhosts connectivity information.
-i *INFORMIXDIR*	Path for the deployed instance. Must be set here or as environment variable (no default path is provided).	
--ifxpass *password*	Specifies the password of the user used to create the service.	Specify the password for user **informix** or the non-root user on the target computer. If you specify a password and the **informix** user does not exist on the target computer, the **informix** user is created with the specified password.
--image_archive *archive_file*, *target_directory*	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.	Valid formats for the *archive_file* are .zip and .tar. If your dbspaces will be in multiple directories, do not use this option. You must create directories for the dbspaces yourself, set the proper ownership and permissions, and extract the snapshot contents into the directories.
--nonroot	Specifies that the instance is owned by the non-root user who runs the command.	UNIX and Linux only.
-o *ONCONFIG*	The onconfig file name. If none is specified here and no ONCONFIG environment variable is set, a new file is created from the onconfig.std file.	
--relocate *path*	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. Windows: If the parent directories have spaces in the paths, place the string of paths after the **--relocate** option in double quotation marks. Example:`--relocate "C:\Program Files\IBM\IBM Informix\14.10\ tmp=D:\IFMXDATA\myserver"` Linux and UNIX: If multiple parent directories are specified and have spaces in the paths, place the string of paths after the **--relocate** option in single quotation marks. Example:`-- relocate '/opt/IBM/informix1210/ tmp=/local/data/myserver; /opt/IBM/informix1210/ dbspace=/local2/data/myserver '`

Element	Purpose	Key considerations
--rootpath *rootpath*	Indicates the location of the root dbspace.	Windows: Default path:*drive:*\ifmxdata*server_name*\root dbs_dat.000 Linux and UNIX: Default path:$INFORMIXDIR/*server_name*/rootdbs
-s *INFORMIXSERVER*	Primary database server name. Must be set either here or in the INFORMIXSERVER environment variable before deployment (no default value is provided).	
--servernum *server_number*	Specifies the server number of the instance.	The number must be an integer from 0 to 255. The default is 0.
--silent	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 **-silent** option and **-verbose** option together, you must also include the **-l** argument.
--snapshot *snapshot_file*	Extracts a compressed snapshot of the file specified as *snapshot_file*.	Windows: Indicate the location of the compressed image as an absolute path. Example: --snapshot C:\IBM\informix\my_archive.zip Linux and UNIX: Example: --snapshot /opt/IBM/informix/ifxdir.tgz
--sqlhosts *sqlhosts_file*	Specifies the name of the SQLHOSTS file.	Default is the value of the INFORMIXDIRSQLHOSTS environment variable, if set.
--start *number*	Starts the database server after the *number* of seconds.	Default is 600.
-UNINSTALL	Removes a snapshot that was originally deployed by the deployment utility.	
--verbose	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 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 ifxdeploy command (Deprecated)
  This version of the **ifxdeploy** command is deprecated but still supported for backward compatibility.

The ifxdeploy command (Deprecated)

This version of the **ifxdeploy** command is deprecated but still supported for backward compatibility.

You can continue to use the following syntax of the **ifxdeploy** command to deploy a snapshot or remove a snapshot that you already deployed.

Syntax

```
>>-ifxdeploy--+-| Deployment Parameters |-----+--------------->< 
              '-| Uninstallation Parameters |-'

Deployment parameters

|--+- -p--password-+--+--------------------------+--+------------------------+-->
   |          (1) |  '- -config--"conf_file_name"-'  +- -file--"full_path_name"-+
   '- -system------'                                 '- -extractcmd--"command"--'

>--+---------------------------------------------+-------->
   +- -silent--+-----+--+----------------------------+-+
   |           '- -y-'  '- -verbose-- -l--"log_file"-' |
   '- -verbose-------------------------------------'

>--+----------------+--+----------------+------------------>
   +- -sqliport--port-+  '- -drdaport--port-'
   |            (2) |
   '- -namedpipe------'

>--+--------------------------+------------------------------>
   '- -servernum--server_number-'

>--+----------------------------------------------------------------------------------+-->
   '- -rootpath--"path"--+-----------------------------------------------------------+-'
                         |                      (3)                                  |
                         '- -relocate------+-path-----------------------------------+-'
                                           | .-;----------------.                   |
                                           | V                  |                   |
                                           +---old_path=new_path-+-----------------+
                                           | .-;--------------------------------.  |
                                           | V                                  |  |
                                           '---old_path,old_offset=new_path,new_offset-+-'

>--+---------------------------+--+---------+---------------->
   |            (4)            | '- -start-'
   '- -installdrive------"drive"-'

>--+----------------+--+-----------+--+---------+-------------->
   '- -autorecommend-'  '- -verbose-'  '- -force-'

>--+----------------+--+-----+---------------------------------|
   '- -l--"log_file"-'  '- -y-'

Uninstallation parameters

|-- -uninstall--"installation_path"--+---------+--------------->
                                     '- -delifx-'

>--+---------------------------------------------+-------->
   +- -silent--+-----+--+----------------------------+-+
   |           '- -y-'  '- -verbose-- -l--"log_file"-' |
   '- -verbose-------------------------------------'

>--+----------------+--+-----+---------------------------------|
   '- -l--"log_file"-'  '- -y-'
```

Notes:

1. **-system** functions on Windows only
2. **-namedpipe** functions on Windows only
3. **-relocate** can run without **-rootpath** option if ROOTPATH is set in an ONCONFIG parameter
4. **-installdrive** functions on Windows only

Command options

Table 1. The **ifxdeploy** command options

Element	Purpose	Key considerations
-autorecommend	Calculates the optimal settings for some onconfig file parameters. This option generates an alternative customized configuration file.	The alternative configuration file might have suggested changes for certain configuration parameters that operate when the server is restarted. Windows: The name of the alternative configuration file is %ONCONFIG%.autorec. Linux and UNIX: The name of the alternative configuration file is $ONCONFIG.autorec.
-config *"conf_file_name"*	Specifies deployment configuration file to run with utility.	
-delifx	Windows: Removes user **informix** and the **Informix-Admin** group. Linux and UNIX: Removes user **informix** and group **informix**.	The **-delifx** option functions when there is only one IBM® Informix® installation on the computer.
-drdaport *port*	Specifies the Distributed Relational Database Architecture™ (DRDA) service port for use with the IBM Common Clients.	The default is 9089.
-file *"full_path_name"*	Extracts a compressed snapshot *only if* the file specified in *"full_path_name"* is in BZIP2, GZIP, TAR, or ZIP format.	Windows: Indicate the location of the compressed image as an absolute path. Example: -file C:\IBM\informix\my_archive.zip Linux and UNIX: Example: -file /opt/IBM/informix/ifxdir.tgz If the snapshot is not in the BZIP2, GZIP, TAR, or ZIP compression format, use the **-extractcmd** option instead.
-extractcmd *"command"*	Extracts a compressed snapshot, regardless of compression format. This option primarily is intended for when you want to extract a snapshot that is *not* in BZIP2, GZIP, TAR, or ZIP format or when you have specialized extraction procedure.	Specify the command and the file that contains the snapshot files. If the snapshot is in BZIP2, GZIP, TAR, or ZIP compression format, then use the **-file** option instead. Example on Windows:`-extractcmd "tar -xf C:\my_archive.tar c:\informix"` Example on Linux and UNIX:`-extractcmd " ( cd /placetoextract ; myextract /opt/IBM/informix/ifxdir) "`
-force	Overwrites the existing instance settings on the target computer.	
-installdrive *"drive"*	This option is for Windows only. Specifies the drive for the directory containing data spaces (typically, this is the IFMXDATA directory).	The default drive is C:. You must have enough space for the installation and databases.
-l *"file_path"*	Sends status messages to a text file.	Indicate the full path and file name for the log file that you want to use. If the file does not exist, it will be created for you.

Element	Purpose	Key considerations
-namedpipe	This option is for Windows only. Sets the default database server network protocol to a named pipe connection.	If you specify both **-namedpipe** and **-sqliport** options, the deployment utility uses only one of them.
-p *password*	Specifies the user **informix** password used to create the service.	Specify the password for user **informix** on the target computer. If you specify a password and the user **informix** does not exist on the target computer, the user will be created and will have the specified password.
-relocate *path*	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. Windows: If the parent directories have spaces in the paths, place the string of paths after the **-relocate** option in double quotation marks. Example:`ifxdeploy -relocate "C:\Program Files\IBM\IBM Informix\11.50\ tmp=D:\IFMXDATA\myserver"` Linux and UNIX: If multiple parent directories are specified and have spaces in the paths, place the string of paths after the **-relocate** option in single quotation marks. Example:`ifxdeploy -relocate '/opt/IBM/informix1150/ tmp=/local/data/myserver; /opt/IBM/informix1150/ dbspace=/local2/data/myserver'`
-rootpath *"path"*	Indicates the location of the root dbspace.	Windows: Default path:*drive:*\ifmxdata*server_name*\rootdbs_dat.000 Linux and UNIX: Default path:$INFORMIXDIR/*server_name*/rootdbs
-servernum *server_number*	Specifies the server number of the instance.	The number must be an integer from 0 to 255. The default is 0.
-silent	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 **-silent** option and **-verbose** option together, you must also include the **-l** argument.
-sqliport *port*	Specifies the SQLHOSTS service port for the server instance.	The default port is 9088. If you specify both **-namedpipe** and **-sqliport**, the deployment utility uses only one of them.

Element	Purpose	Key considerations
-start	Sets the number of seconds for the deployed instance to wait for automatic initialization of shared memory to bring the database server online before returning to the command line.	You must change the setting from zero to a workable number for the **-start** option to function. If you do not pass a value with the **-start** option, the default number of seconds is 600. This option supports the deployment and starting of an instance in a single operation. Using this option, an online operational database is silently deployable.
-system	This option is for Windows only. Creates the service that logs on to the operating system as system user.	
-uninstall *installation_path*	Removes a snapshot that was originally deployed by the deployment utility.	
-verbose	Runs the command in verbose mode.	
-y	Runs the command without prompting for confirmation.	

Usage

Before you use this command, create a snapshot and copy it to the computer where you want to deploy the snapshot. See Creating a snapshot for deployment manually for more information.

Run this command, with options, on the computer where you want to deploy the snapshot. When you specify a value for a parameter, if the value contains a space, enclose the value in double quotation marks. You must run the command either as an Administrator user or as the **root** user.

You must provide the location of the snapshot by using the **-extractcmd** option, the **-file** option, or by setting the SNAPSHOT parameter in the ifxdeploy.conf file.

If the snapshot is in a different compressed format, you must extract it. Use the **-file** option to extract a snapshot if it is in BZIP2, GZIP, TAR, or ZIP compression format. If the snapshot is in a different compression format, use the **-extractcmd** option with a customized command or script.

If you included data spaces in your snapshot in compressed format, you must extract the data spaces before you run the **ifxdeploy** command. Alternatively, you can write a shell script to use with the **-extractcmd** option that specifies how to extract both the server snapshot and the data spaces snapshot.

Use the **-force** parameter to deploy a snapshot to multiple computers and override the database server instance settings on the target computer.

On Windows, the **ifxdeploy** command deploys a snapshot of Informix, which is not visible. If you do not want to use Informix, use the **ifxdeploy** utility to uninstall it.

Examples of deployment utility usage on Windows

Deploying the instance in silent mode
> In this example, the user **informix** is created on the target server and has the password `mypassw0rd`. The command will run in silent mode, without prompting for confirmation. The snapshot is extracted from the my_archive.zip file by 7-zip software if it is installed on the operating system.

> ```
> ifxdeploy.exe -p mypassw0rd -y -silent
>  -file C:\IBM\informix\my_archive.zip
> ```

Deploying the instance as local system user and specifying an extraction method
> In this example, the snapshot is extracted from the C:\my_archive.tar file on the target computer.

> ```
> ifxdeploy.exe -system -extractcmd
>  "tar -xf C:\my_archive.tar c:\informix"
> ```

Because the **tar -xf** part of the argument to the **-extractcmd** option contains a space, the argument and the paths after it are enclosed in double quotation marks.

Creating a new server instance

In this example, the SQLHOSTS port number is 9090, the server number for the instance is 2, and a full path is specified for the log file.

```
ifxdeploy.exe -silent -y -sqliport 9090
 -servernum 2 -l C:\my_log.txt
```

Creating an instance as local system user with DRDA enabled

In this example, the SQLHOSTS port is 9090, the DRDA port is 9096, and the command will run in verbose mode. The snapshot used here is not in a compressed file format, so neither the **-file** option nor the **-extractcmd** option are required.

```
ifxdeploy.exe -system -y -sqliport 9090
 -drdaport 9096 -verbose
```

Dynamic relocation of chunks to single parent directory

In this example, the chunks are relocated to a single parent directory and indicate location of the root dbspace:

```
ifxdeploy -rootpath D:\IFMX\ex1\rootdbs.001
 -relocate D:\IFMX\ex2
```

Chunk relocation to multiple paths

In this example, the location of the root dbspace has been specified with the ROOTPATH configuration parameter, so the command is not required to run with the **-rootpath** option. Each mapping between the old path and the new path is separated by a semicolon.

```
ifxdeploy -relocate C:\IFMXDATA=D:\IFMXDATA;
C:\IFMXLOGSPACE=E:\IFMXLOGSPACE
```

Chunk relocation to multiple paths with new offsets

In this example, the location of the root dbspace has been specified with the ROOTPATH configuration parameter, so the command is not required to run with the **-rootpath** option. The mapping of the old paths and the new paths includes offset values (in KB), which are indicated after the commas.

```
ifxdeploy -relocate C:\IFMXDATA,0=D:\IFMXDATA,4;
C:\IFMXDATAB,3=D:\IFMXDATAB,5
```

Automatic startup and initialization of the deployed instance

In this example, the command runs as follows:

- Does not prompt for confirmation (the **-y** option).
- Extracts a snapshot located at C:\work\ifxdir.zip.
- Sets the server number of the deployed instance to 2 (the **-servernum** option).
- Directs the deployed instance to attempt automatic initialization and startup (the **-start** option) within 300 seconds. If initialization is not complete after 300 seconds, the computer returns to the command line.

```
ifxdeploy -file C:\work\ifxdir.zip
 -servernum 2 -start 300 -y
```

Automatic startup, initialization of the deployed instance, and creation of a customized configuration file

In this example, the command runs as follows:

- Does not prompt for confirmation (the **-y** option).
- Extracts a snapshot located at C:\work\ifxdir.zip.
- Sets the server number of the deployed instance to 2 (the **-servernum** option).
- Directs the deployed instance to attempt automatic initialization and startup (the **-start** option) within 300 seconds. If initialization is not complete after 300 seconds, the computer returns to the command line.
- Creates an alternative configuration file with optimal settings.

```
ifxdeploy -file C:\work\ifxdir.zip -servernum 2
 -start 300 -autorecommend -y
```

Automatic startup and initialization of the deployed instance with chunk relocation to multiple paths

In this example, the command runs as follows:

- Runs in verbose mode (the **-verbose** option) and does not prompt for confirmation (the **-y** option).
- Extracts a snapshot located at C:\work\ifxdir.zip.

- Directs the deployment utility to dynamically relocate chunks to a single parent directory (the **-relocate** option) and sets the location of the root dbspace (the **-rootpath** option).
- Sets the server number of the deployed instance to 2 (the **-servernum** option).
- Directs the deployed instance to attempt automatic initialization and startup (the **-start** option). Because no value is set with the **-start** option, the computer waits the default time of 600 seconds before returning to the command line.

```
ifxdeploy -file C:\work\ifxdir.zip -verbose -servernum 2
-relocate C:\work\chunks\instance2
-rootpath C:\work\chunks\instance2\rootdbs -start -y
```

Examples of deployment utility usage on Linux and UNIX

Extracting a snapshot saved as a .tgz file with Gzip and other options
In this example, the deployment utility extracts a snapshot saved as /opt/IBM/informix/ifxdir.tgz by using Gzip. The command will run in verbose mode, create the log file /tmp/mylog, and not prompt for confirmation.

```
ifxdeploy -file /opt/IBM/informix/ifxdir.tgz
 -l /tmp/mylog -verbose -y
```

Specifying SQLHOSTS settings and the server number for the deployed instance
In this example, the files are already in place and you are creating a new instance by specifying port numbers for DRDA and the service port, as well as setting the server number for the instance. The snapshot is not compressed, so neither the **-extractcmd** option nor the **-file** option are required. The INFORMIXDIR and INFORMIXSERVER environment variables have been updated to new values as necessary.

```
ifxdeploy -sqliport 9093 -drdaport 9094 -servernum 3
```

Dynamic relocation of chunks to single parent directory
In this example, the chunks are relocated to a single parent directory and indicate location of the root dbspace:

```
ifxdeploy -rootpath /opt/ibm/IDS/exa/space
 -relocate /opt/ibm/IDS/exa/space2
```

Chunk relocation to multiple paths
In this example, the location of the root dbspace has been specified with the ROOTPATH configuration parameter, so the command is not required to run with the **-rootpath** option. Each mapping between the old path and the new path is separated by a semicolon.

```
ifxdeploy -relocate /opt/IBM/ex3=/idsb/myserver;
/opt/IBM/ex4=/idslogs/myserver
```

Chunk relocation to multiple paths with new offsets
In this example, the location of the root dbspace has been specified with the ROOTPATH configuration parameter, so the command is not required to run with the **-rootpath** option. The mapping of the old paths and the new paths includes offset values (in KB), which are indicated after the commas.

```
ifxdeploy -relocate /opt/IBM/dbspaces,
0=/lv1/data,4;/opt/IBM,2=/ids/myserver,6
```

Automatic startup and initialization of the deployed instance
In this example, the command runs as follows:

- Does not prompt for confirmation (the **-y** option).
- Extracts a snapshot located at /work/ifxdir.tgz (the **-file** option).
- Sets the server number of the deployed instance to 2 (the **-servernum** option).
- Directs the deployed instance to attempt automatic initialization and startup (the **-start** option) within 300 seconds. If initialization is not complete after 300 seconds, the computer returns to the command line.

```
ifxdeploy -file /work/ifxdir.tgz
 -servernum 2 -start 300 -y
```

Automatic startup, initialization of the deployed instance, and creation of a customized configuration file
In this example, the command runs as follows:

- Does not prompt for confirmation (the **-y** option).
- Extracts a snapshot located at /work/ifxdir.tgz (the **-file** option).
- Sets the server number of the deployed instance to 2 (the **-servernum** option).

- Directs the deployed instance to attempt automatic initialization and startup (the **-start** option) within 300 seconds. If initialization is not complete after 300 seconds, the computer returns to the command line.
- Creates an alternative configuration file with optimal settings.

```
ifxdeploy -file /work/ifxdir.tgz
 -servernum 2 -start 300 -autorecommend -y
```

Automatic startup and initialization of the deployed instance with chunk relocation to multiple paths
In this example, the command runs as follows:

- Runs in verbose mode (the **-verbose** option) and does not prompt for confirmation (the **-y** option).
- Extracts a snapshot located at /work/ifxdir.tgz.
- Directs the deployment utility to dynamically relocate chunks to a single parent directory (the **-relocate** option) and sets the location of the root dbspace (**-rootpath** option).
- Sets the server number of the deployed instance to 2 (the **-servernum** option).
- Directs the deployed instance to attempt automatic initialization and startup (the **-start** option). Because no value is set with the **-start** option, the computer waits the default time of 600 seconds before returning to the command line.

```
ifxdeploy -file /work/ifxdir.tgz -verbose -servernum 2
-relocate /work/chunks/instance2
-rootpath /work/chunks/instance2/rootdbs -start -y
```

- The ifxdeploy.conf file (Deprecated)
  The ifxdeploy.conf file is a text-file template in which you can configure an instance snapshot before deploying it with the deprecated version of the **ifxdeploy** command.

The ifxdeploy.conf file (Deprecated)

The ifxdeploy.conf file is a text-file template in which you can configure an instance snapshot before deploying it with the deprecated version of the **ifxdeploy** command.

Purpose

The ifxdeploy.conf file is in $INFORMIXDIR/etc/.

The ifxdeploy.conf file allows for more dynamic customization than the command-line options because in the file you can specify any configuration parameter values, set key environment variables, and create multiple database server aliases. You can save and reuse the file to deploy instances to other locations. The file supports the same functionality as the **ifxdeploy** command options, but also additional functionality that is useful for embedding IBM® Informix® when you are deploying a snapshot in multiple locations that require minimal or no modification in instance setup. See Planning for deploying Informix for a description of the major embeddability features of the ifxdeploy.conf file.

Guidelines for Usage

To apply configuration settings of the ifxdeploy.conf file to an instance you are deploying, enter the file name as an argument to the `-config` option when you run the **ifxdeploy** command.

If you set different values for the same instance in the ifxdeploy.conf file and as an option to the **ifxdeploy** command option, the deployment utility uses the value specified on the command line. For example, if you specify `-p mypassword1` on the command line but set the INFORMIXPASSWORD parameter to `mypassword2` in the ifxdeploy.conf file, the deployed instance requires `mypassword1` for authentication.

Parameters

The ifxdeploy.conf file is value pair based. If there is a parameter with a default value that you want to change, provide the value in an uncommented line.

Attention: The values of parameters that are set on the **ifxdeploy** command line overwrite the values of the same parameters in the ifxdeploy.conf file.

The following table explains the parameters in the same order that they appear in the configuration template file.

Table 1. The ifxdeploy.conf file parameters

Parameter	Description	Example of value setting (uncommented line)
INFORMIXSERVER	Primary database server name. Must be set either here or as environment variable before deployment (no default value is provided).	`INFORMIXSERVER deploy3`
PROTOCOL1	Primary network protocol. This parameter is equivalent to the connection type in the sqlhosts information.	`PROTOCOL1 olscoctcp`
SQLIPORT	SQLHOSTS service port for the server instance (not required for onipcnmp). The range of permissible values is from 1 to 65536.	`SQLIPORT 9088`
DRDAPORT	The Distributed Relational Database Architecture™ (DRDA) service port for use with the IBM Common Clients. The range of permissible values is from 1 to 65536.	`DRDAPORT 9089`
SERVERNUM	The server number. Corresponds to the SERVERNUM configuration parameter. The range of permissible values is from 0 to 255.	`SERVERNUM 100`
INFORMIXSQLHOSTS	**Linux and UNIX**: Full path to sqlhosts file for the instance to use. Windows: Pointer to remote computer containing SQLHOSTS registry settings that the deployed instance is to use.	
BEGIN ALIAS . . . END ALIAS	Specifies new database server aliases and related SQLHOSTS connectivity settings for the deployed instance. The optional OPTIONS line sets an SQLHOSTS parameter value. In the example, `b=32767` sets buffers.	**BEGIN ALIAS** **SERVERNAME alias1** **PROTOCOL drsoctcp** **PORT 9091** **OPTIONS b=32767** **END ALIAS**
INFORMIXDIR	Path for the deployed instance. Must be set here or as environment variable (no default path is provided).	Windows: C:\tmp\informix **UNIX andLinux**: /tmp/informix
ONCONFIG	The onconfig file name. If none is specified here and no ONCONFIG environment variable is set, a new file is created from onconfig.std.	`onconfig.sample`
START	Set this to the number of seconds for the deployed instance to wait for database server initialization to complete before returning to the command line. Use this parameter in an embeddability environment that requires noninteractive startup of . The setting of 0 directs the deployed instance to not attempt automatic startup. If you do not specify a value with the `-start` option, the target computer waits the default time of 600 seconds before returning to the command line.	`START 90`

Parameter	Description	Example of value setting (uncommented line)
SNAPSHOT	This parameter can only be used if you are deploying an instance from a snapshot compressed as a .tgz file on UNIX or Linux or a .zip file on Windows. Indicates the location of a supported, compressed archive type containing the snapshot. This parameter is equivalent to the `-file` command-line option.	
RELOCATE	Set chunk paths for the deployed instance. You can indicate a parent directory for all chunk path names or map them individually to separate parent directories. You can also deploy the chunks with specific offset values. Offset values are in KB. You must have the appropriate ownership and permissions to update the directories.	Example 1: Relocate all chunk paths to one directory: Windows: C:\example1 UNIX and Linux: /example1 Example 2: Select individual chunk paths and specify the directories where the paths are relocated: Windows: C:\ex2=C:\ex3; C:\ex4=C:\ex5 UNIX and Linux: /ex2=/ex3;/ex4=C:ex5 Example 3: Change multiple chunk paths and offsets: Windows: C:\ex6,10=C:\ex7,100; C:\ex8,20=C:\ex9,200 UNIX and Linux: /ex6,10=/ex7,100; /ex8,20=/ex9,200
INFORMIXPASSWORD	Password for user **informix** on the target computer. Sets password to what you enter as a value if no user **informix** exists on the computer. If you do not supply a password, you are prompted to enter a password by the **ifxdeploy** utility.	`INFORMIXPASSWORD` *password*
SYSTEM	Specifies whether the deployed instance logs on to Windows as a local system user or as user **informix**. This parameter is for Windows only.	0 (default). logs on as user **informix**. 1. logs on as the Local System user and adds the current user to the **Informix-Admin** group.
LOGFILE	Sets the full path name for the log file of the deployment utility's errors and messages.	**Windows:** LOGFILE c:\my_log.txt **UNIX andLinux**: LOGFILE /tmp/mylog
LOGLEVEL	Sets amount of information to write to log. See the ifxdeploy.conf file for permissible values.	`LOGLEVEL 5`
SILENT	Sets whether the utility displays console output while it is running.	`SILENT 1` (no console output) `SILENT 0` (displays console output)
FORCE	Overwrites existing environment variable and onconfig file settings of the target computer.	`FORCE 0` (Does not overwrite) `FORCE 1` (Overwrites existing settings)
INSTALLDRIVE This parameter is for Windows only.	Specifies the directory in which the deployed instance's dbspaces will be created on Windows.	`INSTALLDRIVE C`

Parameter	Description	Example of value setting (uncommented line)
ROOTPATH	Sets the location of the root dbspace.	Windows: Default path: *drive:*\ifmxdata*server_name*\rootdbs_dat.000 **UNIX and Linux** Default path: $INFORMIXDIR/*server_name*/rootdbs
WIN6432 This parameter is for Windows only.	Set this to 1 if installing 32-bit on 64-bit Windows.	`WIN6432 1`

BEGIN ONCONFIG . . . END ONCONFIG	Specify values for any configuration file parameters. Enter each parameter and value exactly as they would be entered in the onconfig file.
	
	Can be used to overwrite specific onconfig file parameter values or instead of providing an onconfig file.
		`BEGIN ONCONFIG`
		`LOCKS 10000`
		`END ONCONFIG`

BEGIN AUTORECOMMEND . . . END AUTORECOMMEND	Specifies the following parameters that are used to generate recommended values in the onconfig file:
	
	MAXCPUS
	    Maximum number of processors and cores that the database server can use.
	    Default: 1
	    Minimum value: 1
	    Maximum value: none
	MAXDISK
	    Maximum amount of disk space that the database server can use.
	    Default: 2048 MB
	    Minimum value: 256 MB
	    Maximum value: none
	MAXMEM
	    Maximum amount of memory that the database server can use.
	    Default: 512 MB
	MAXUSERS
	    Maximum number of expected online transaction processing (OLTP) application users.
	    Default: 32
	MAXDSUSERS
	    Maximum number of expected decision-support system (DSS) application users.
	    Default: 4
	RTO_SERVER_RESTART
	    Recovery time objective for a server restart.
	    Default: 60 seconds
	    Minimum value: 60 seconds
	    Maximum value: 1800 seconds
		`BEGIN AUTORECOMMEND`
		`MAXCPUS 1`
		`MAXDISK 2048`
		`MAXMEM 512`
		`MAXUSERS 32`
		`MAXDSUSERS 4`
		`RTO_SERVER_RESTART 60`
		`END AUTORECOMMEND`

CLONE	Set this to 1 to deploy a clone of a source server. Specify the information for the source server in the parameters nested between the BEGIN CLONE and END CLONE statements of the ifxdeploy.conf file.	`1`

Parameter	Description	Example of value setting (uncommented line)
SOURCESERVER	Specifies the name of the source server.	`SOURCESERVER clone 3`
SOURCEIPADDR	Specifies the IP address of the source server.	
SOURCEPORT	Specifies the port number of the source server's listener thread.	
CLONEIPADDR	Specifies the IP address of the clone server.	
DISPOSITION	Specifies the final disposition of the clone server.	`DISPOSITION RSS`
TARGETSIZE	Specifies the size of clone server.	`TARGETSIZE medium`
USELOCAL	Indicates to use local configuration after merging with source configuration.	`USELOCAL 1`
TRUSTED	Specifies that the user is trusted and that it is not necessary to obtain a user ID and password to access the server.	`TRUSTED 1`
USERNAME	Specifies the name of user for connecting to source server	`USERNAME informix`
PASSWORD	Specifies the password of the name specified in the USERNAME parameter.	`PASSWORD password`

Related information:
The sqlhosts information

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--+-----------------------+--+---------------------+--+---------------------+->< 
                 '- -source--sqlhostvalue-'  '- -target--sqlhostvalue-'  '- -servers--serverlist-'
```

Command options

Table 1. Options for the syncsqlhosts command

Option	Purpose	Key considerations
`-source sqlhostvalue`	Specifies the source machine to copy connection information from. `sqlhostvalue` is either a \\machine name or an sqlhosts file.	If you do not specify a -source option, the local registry is used.
`-target sqlhostvalue`	Specifies the target machine to copy connection information to. `sqlhostvalue` is either a \\machine name or an sqlhosts file.	If you do not specify a -target option, the local registry is used.
`-servers serverlist`	Specifies the servers to copy connection information from. `serverlist` is a list of Informix® Server entries that are separated by spaces that are copied from the source to the target machine.	If you do not specify a -servers 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 (").

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 concepts:
Synchronize connectivity information between servers and clients on Windows

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.

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

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.

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 for details about the command syntax.
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
```

Deploying Informix and client products with RPM-based installation (Linux)

You can create RedHat Package Manager (RPM) images of a customized IBM® Informix® product installation to deploy the products with a silent installation. You can use RPM images to deploy the database server and client products in embedded applications. After you install the database server with an RPM image, you must manually initialize the database server.

See the machine notes for the specific Linux operating systems that are supported.

Creating RPM images

Before you install products, complete the following tasks:

- Check that RPM is installed on your computer.
- Log in as the appropriate user to be the owner of the database server. For more information, see Installation owner.

To create RPM images:

Run an interactive installation of and associated client products and choose the RPM installation method. For instructions, see Running an interactive installation.. The following RPM images are created in subdirectories under the target installation directory:

- RPMS/gskcrypt*gskit_version*.linux.*processor*.rpm
- RPMS/gskssl*gskit_version*.linux.*processor*.rpm
- *processor*/informix-*informix_version*.*processor*.rmp

The *gskit_version* is the version of IBM Global Security Kit (GSKit), for example: 64-8.0.50.20. The *processor* is the processor architecture of the computer, for example, x86_64. The *informix_version* is the database server version, for example: 12.10.FC5-011.70.FC8-0.
For example, if you install version 12.10.FC5 on a Linux x86_64 computer, you generate the following RPM images:

```
RPMS/gskcrypt64-8.0.50.20.linux.x86_64.rpm
RPMS/gskssl64-8.0.50.20.linux.x86_64.rpm
x86_64/informix-12.10.FC5-0.x86_64.rpm
```

Installing RPM images

To install RPM images on a target computer:

1. Log in with root privileges. Root-level privileges are required to deploy the RPM image, even if you are deploying a non-root installation.
2. Copy the three RPM images to a local directory.
3. Run the following command for each RPM image:

   ```
   rpm -i image_file_name --prefix absolute_installation_path
   ```

   If you do not specify an installation path with the `--prefix` option, then the default installation location is /opt/IBM/Informix.

   Tip: Record the image file name after you complete a deployment. If you uninstall the product, you need to use the image file name.
4. If you are deploying a non-root installation of the database server, change the ownership of the installation directory to the non-root owner.
5. Initialize the database server. For instructions, see Creating a database server after installation.

Removing an installation

To remove an installation with RPM:

1. Log in as the root user.
2. Run the following command for each of the three RPM images:

```
rpm -e image_file_name
```

Informix embeddability deployment tutorial

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

The IBM Informix 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 Linux shell script and the Windows batch script for this silent deployment example are posted separately on the Technote at http://www.ibm.com/support/docview.wss?uid=swg21446737.

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

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

- Introducing the Informix deployment assistant
  The deployment assistant is used to create snapshots of IBM Informix instances and their associated dbspaces, which can be later deployed on multiple target computers using the deployment utility.
- Introducing the Informix deployment utility
  You can use the deployment utility to deploy snapshots of pre-configured instances and related dbspaces onto target computers.
- Automating deployments of reduced-footprint Informix on Linux
  This section contains information about installing Informix, creating an instance, using the deployment assistant to create a snapshot of the installation, and silently deploying the snapshots on Linux.
- Automating deployments of reduced-footprint Informix on Windows
  This section contains information about installing Informix, creating an instance, using the deployment assistant to create a snapshot of the installation, and silently deploying the snapshots on Windows.

Related concepts:
Planning for deploying Informix

Introducing the Informix deployment assistant

The deployment assistant is used to create snapshots of IBM® Informix® 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 installation by displaying file associations at the time of packaging.

Using the deployment assistant

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

```
ifxdeployassist
```

Related tasks:
Creating a snapshot with the deployment assistant

Introducing the Informix deployment utility

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

You must install IBM® Informix® 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 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 server at $INFORMIXDIR/bin on Linux and UNIX and at %INFORMIXDIR%\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 $INFORMIXDIR/bin on Linux and UNIX and at %INFORMIXDIR%\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 server at $INFORMIXDIR/etc on Linux and UNIX and at %INFORMIXDIR%\etc on Windows.
- You can also specify instance-specific information for the instance that is getting deployed in the configuration file:
    - INFORMIXSERVER sets the primary server name.
    - SERVERNUM sets the primary server port number.
    - INFORMIXSQLHOSTS sets the value for the INFORMIXSQLHOSTS 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

```
INFORMIXSERVER  demo_on
PROTOCOL1 onsoctcp
PORT1 9088
SERVERNUM 1

BEGIN ALIAS
SERVERNAME alias1
PROTOCOL drsoctcp
PORT 9091
END ALIAS

INFORMIXDIR "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 tasks:
Deploying Informix software

Automating deployments of reduced-footprint Informix on Linux

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

- Perform Informix installation and create an instance on Linux
  The first task in the embeddability tutorial is to complete an installation and create the IBM Informix instance.
- Creating a snapshot for deployment on Linux
  The second task in the embeddability tutorial is to create a snapshot of the IBM Informix install binary files and the dbspaces on the template computer by using the deployment assistant.
- Silently deploy snapshots on Linux
  The third task of the embeddability tutorial is to silently deploy the IBM Informix snapshot.

Perform Informix installation and create an instance on Linux

The first task in the embeddability tutorial is to complete an installation and create the IBM® Informix® instance.

Install 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 . 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 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:

- 14.10 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 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 IBM® Informix® install binary files and the dbspaces on the template computer by using the deployment assistant.

Prerequisites:

- The 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 $INFORMIXDIR/bin/ifxdeploy and $INFORMIXDIR/etc/ifxdeploy.conf outside of the $INFORMIXDIR directory. (You will need to have these copies outside of $INFORMIXDIR when you deploy the snapshot on the target computer.)
- The instance from which you want to create the snapshot is running.

1. Start a terminal window.
2. Set the **INFORMIXDIR** and **INFORMIXSERVER** environment variables.
   For example:

   ```
   export INFORMIXDIR=/opt/IBM/informix
   export INFORMIXSERVER=demo_on
   ```

3. Start the deployment assistant with the following commands:

   ```
   cd $INFORMIXDIR/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 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 $INFORMIXDIR/bin/ifxdeploy file, and the $INFORMIXDIR/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 concepts:
Easy deployment

Copyright© 2020 HCL Technologies Limited

Silently deploy snapshots on Linux

The third task of the embeddability tutorial is to silently deploy the IBM® Informix® snapshot.

Deploying the 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.

Important: Refer to the shell script posted on the Technote at http://www.ibm.com/support/docview.wss?uid=swg21446737. The script contains comments that provide information about the purpose of each step. You can edit and customize the script to fit any deployment needs.
Prerequisites:

* You must be logged in to the target computer as root user.
* You must have a snapshot of the instance and any associated dbspaces that was created with the deployment assistant.
* The INFORMIXDIR 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:
  * 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 INFORMIXSERVER, ONCONFIG, and INFORMIXSQLHOSTS 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 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.

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.

- To silently deploy the 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 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 base server instance, deploys associated dbspaces to `/data/IBM/informix/demo/server`, and starts the instance.

See the informixtemp/ifxdeploy.log file for messages that are logged by the deployment utility. If the 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 instance.

Automating deployments of reduced-footprint Informix on Windows

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

- Perform Informix installation and create an instance on Windows
  The first task in the embeddability tutorial is to complete an installation and create the IBM Informix instance.
- Creating a snapshot for deployment on Windows
  The second task in the embeddability tutorial is to create a snapshot of the IBM Informix install binary files and the dbspaces on the template computer by using the deployment assistant.
- Silently deploy Informix on Windows
  The third task of the embeddability tutorial is to silently deploy the IBM Informix snapshot.

Perform Informix installation and create an instance on Windows

The first task in the embeddability tutorial is to complete an installation and create the IBM® Informix® instance.

Install 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 . 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 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:

- 14.10 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 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 IBM® Informix® install binary files and the dbspaces on the template computer by using the deployment assistant.

Prerequisites:

- The 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 %INFORMIXDIR%\bin\ifxdeploy.exe and `%INFORMIXDIR%\etc\ifxdeploy.conf` files in a folder outside %INFORMIXDIR%. You will need these files when you deploy on the target computer.
- The instance from which you want to create the snapshot is running.

1. Start a command window.
2. Set the INFORMIXDIR and INFORMIXSERVER environment variables.
   For example, for the embeddability tutorial, specify:

   ```
   set INFORMIXDIR=C:\Program Files\IBM\Informix
   set INFORMIXSERVER=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 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 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 %INFORMIXDIR%\bin\ifxdeploy.exe file and the `%INFORMIXDIR%\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 Informix on Windows

The third task of the embeddability tutorial is to silently deploy the IBM® Informix® snapshot.

Deploying the 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.

Important: Refer to the batch script posted on the Technote at http://www.ibm.com/support/docview.wss?uid=swg21446737. The script contains comments that provide information about the purpose of each step. You can edit and customize the script to fit any deployment needs.
Prerequisites:

- You must be logged in to the target computer as an Administrator user.
- You must have a snapshot of the 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 INFORMIXDIR 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:
  - 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 INFORMIXSERVER 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 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.

   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.

   - To silently deploy the 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 instance, relocates dbspaces from C:\IFMXDATA\demo_on to C:\IFMXDATANEW\demo_onand starts the instance.

   - To silently deploy the 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 instance, deploys dbspaces to C:\IFMXDATA\demo_on, and starts the instance.

See the C:\informixtemp\ifxdeploy.log file for messages that are logged by the deployment utility. If the 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 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.
- 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.
- 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.
- 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.

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

Related information:
Compression
B-tree index compression
Automatic space management
Partition defragmentation

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.
- 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.
- 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.
- 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.
- 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.

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.

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.

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).

Related concepts:

Related tasks:

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 1. 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
online.log	**online_log_rotate** task	3:00 A.M. every 30 days You can change the frequency, which is specified in the MAX_MSGPATH_VERSIONS parameter of the **ph_threshold** table.
bar_act_log	**bar_act_log_rotate** task	3:00 A.M. every 30 days You can change frequency, which is specified in the MAX_BAR_ACT_LOG_VERSIONS parameter of the **ph_threshold** table.
bar_debug_log	**bar_debug_log_rotate** task	3:00 A.M. every 30 days You can change frequency, which is specified in the MAX_BAR_DEBUG_LOG_VERSIONS parameter of the **ph_threshold** 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.

Related tasks:

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.

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";
```

You can also disable the running of message lot 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";
```

Related concepts:
Tasks that automatically rotate message log files
Related tasks:
Enabling and disabling the tasks that automatically rotate message log files
Configuring the maximum number of message log files to retain
Reducing the size of message log files

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.

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";
```

Related concepts:
Tasks that automatically rotate message log files
Related tasks:
Enabling and disabling the tasks that automatically rotate message log files
Configuring the frequency of automatic rotation of message log files
Reducing the size of message log files

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.

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.logfiles, 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.logfile, specify:

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

Related concepts:
Tasks that automatically rotate message log files
Related tasks:
Enabling and disabling the tasks that automatically rotate message log files
Configuring the frequency of automatic rotation of message log files
Configuring the maximum number of message log files to retain
Related information:
message log rotate argument: Rotate the message log file (SQL administration API)
message log truncate argument: Delete the contents of a message log file (SQL administration API)
message log delete argument: Delete a message log file (SQL administration API)

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
Reserve memory for critical activities

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
scheduler lmm enable argument: Specify automatic low memory management settings (SQL administration API)
onstat -g lmm command: Print low memory management information
Configure the server response when memory is critically low