Installing DB2 servers without root privileges (Linux and UNIX)

Contents

Non-root installation overview (Linux and UNIX)

The DB2® installer automatically creates and configures a non-root instance during a non-root installation. As a non-root user, you can customize the configuration of the non-root instance during the installation. You can also use and maintain the installed DB2 database product without root privileges.

The non-root installation of a DB2 database product has one DB2 instance with most features enabled by default.

A non-root installation can be attractive for many groups, such as:

- Enterprises that have thousands of workstations and users who want to install a DB2 database product without using a system administrator's time
- Application developers who are not typically system administrators but use DB2 database products to develop applications
- Independent software vendors (ISVs) who develop software that does not require root user authority yet embeds a DB2 database product

Although non-root installations have most of the function of root installations, there are some differences and limitations. You can lift some of the limitations by having a root user run the **db2rfe** command.

Differences between root installations and non-root installations

In addition to a few limitations, the directory structure of a non-root installation is slightly different from the directory structure of a root installation.

During a root installation, subdirectories and files for the DB2 database product are created in a directory of the root user's choosing.

Unlike root users, non-root users cannot choose where DB2 database products are installed. Non-root installations are always placed in the *$HOME*/sqllib directory, where *$HOME* represents the non-root user's home directory. The layout of the subdirectories within the sqllib directory of a non-root is similar to that of a root installation.

For root installations, multiple instances can be created. Instance ownership is associated with the user ID under which the instance was created.

Non-root installations can have only one DB2 instance. The non-root installation directory contains all of the DB2 database product files and instance files with no soft links.

The following table summarizes the differences between root installations and non-root installations.

Table 1. Differences between root installations and non-root installations

Criteria	Root installations	Non-root installations
User can select installation directory	Yes	No. DB2 database products are installed under the user's home directory.

Table 1. Differences between root installations and non-root installations (continued)

Criteria	Root installations	Non-root installations
Number of DB2 instances allowed	Multiple	One
Files that are deployed during installation	Program files only. Instances must be created after installation.	Program files and instance files. The DB2 database product is ready for use immediately after installation.

Limitations of non-root installations

In addition to the differences between root installations and non-root installations, there are several limitations on non-root installations. This topic discusses the limitations to help you decide whether you want to use a non-root installation.

Product limitations

Some DB2 database products are not supported in non-root installations:

- DB2 Net Search Extender
- Locally installed *DB2 Information Center*

  Note: The locally installed *DB2 Information Center* is not supported in non-root installations because it requires root user authority to start the daemon. However, a non-root-installation DB2 instance can be configured to use a locally installed *DB2 Information Center* if it is installed on the same computer.

Features and tools limitations

The following features and tools are not available in non-root installations:

- The DB2 Administration Server (DAS) and its associated commands: **dascrt**, **dasdrop**, **daslist**, **dasmigr**, and **dasupdt**
- The ability for the **db2governor** to increase priority is not supported
- Automatic starting of non-root DB2 instances at system reboot is not supported

Health monitor limitations

The following health monitor features are not supported in non-root installations:

- Running script or task actions on alert occurrences
- Sending alert notifications

Partitioned database limitation

Only single-partition databases are supported in non-root installations. You cannot add extra database partitions.

Listing DB2 database products

The output that is produced by the **db2ls** command, when run as a non-root user, is different than the output produced when run as a root user. For details, refer to the **db2ls** command topic.

DB2 copies

Each non-root user can have only one copy of a DB2 database product installed.

DB2 instance limitation

In non-root installations, one DB2 instance is created during installation. Extra instances cannot be created.

DB2 instance actions can be performed only by the instance owner

Root installations and non-root installations can coexist on the same computer in different installation paths. However, a non-root instance can be updated, or dropped (by using the **db2_deinstall** command), only by the non-root user who owns the non-root instance.

A DB2 instance that is created by a user with root user authority can be updated or dropped only by a user with root user authority.

Upgrading limitation

Root instances cannot be upgraded to a non-root instance.

Manual kernel parameter updates required

Automatic Linux kernel parameter modification is not supported for non-root installations. Kernel parameters in non-root installations must be updated manually as described in the "Modifying Kernel Parameters (Linux)" topic.

Post-installation actions can be performed only by the DB2 instance owner

Root installations and non-root installations can coexist on the same computer. However, only the original non-root user who installed the DB2 database product can perform subsequent actions such as:

- Applying fix packs
- Adding features
- Installing add-on products

Adjusting `ulimit` values

The **ulimit** command on UNIX and Linux operating systems sets or reports user resource limits, such as data and stack limits. For root instances, the database server dynamically updates required **ulimit** settings without changing the permanent settings. However, for non-root instances, the **ulimit** settings can be checked only during installation. A warning message is displayed if the settings are inadequate. Root user authority is required to change the **ulimit** settings.

Limitations that can be overcome by running db2rfe

There are further limitations on non-root installations, which can be overcome by running the **db2rfe** command. The following features and abilities are initially unavailable in non-root installations:

- Operating system-based authentication
- High Availability (HA) feature
- The ability to reserve service names in the `/etc/services` file
- The ability to increase user data limits (**ulimit**). This ability applies only to AIX®. On other operating systems, user data limits must be increased manually.

Run the “Enable root features for non-root installation” command (**db2rfe**) to enable these features and abilities. Running the **db2rfe** command is optional, and must be run by a user with root user authority.

Authentication type in non-root installations

Operating system-based authentication is the default authentication type for DB2 database products. Since non-root installations do not support operating system-based authentication, if you choose not to run the **db2rfe** command after you install your DB2 database product as a non-root user, then you must manually set the authentication type. You can do so by updating the following parameters in the database manager configuration (dbm cfg) file:

- **clnt_pw_plugin** (Client userid-password plug-in configuration parameter)
- **group_plugin** (Group plug-in configuration parameter)
- **srvcon_pw_plugin** (Userid-password plug-in for incoming connections at the server configuration parameter)

Installing DB2 database servers as a non-root user

Most DB2 database products can be installed as a non-root user.

Before you begin

Before you install any DB2 database product as a non-root user, be aware of the differences between root installations and non-root installations, and the limitations of non-root installations. For more information on non-root installation, see "Non-root installation overview (Linux and UNIX)".

Prerequisites for installing a DB2 database product as a non-root user:

- You must be able to mount the installation DVD, or have it mounted for you.
- You must have a valid user ID that can be used as the owner of a DB2 instance.

  User IDs have the following restrictions and requirements:
  - Must have a primary group other than guests, admins, users, and local
  - Can include lowercase letters (a-z), numbers (0-9), and the underscore character (_)
  - Cannot be longer than 8 characters
  - Cannot begin with IBM, SYS, SQL, or a number
  - Cannot be a DB2 reserved word (USERS, ADMINS, GUESTS, PUBLIC, or LOCAL), or an SQL reserved word
  - Cannot use any User IDs with root privilege for the DB2 instance ID, DAS ID, or fenced ID
  - Cannot include accented characters
  - If existing user IDs are specified instead of creating new user IDs, make sure that the user IDs are not locked and do not have expired passwords
- The hardware and software prerequisites that exist for a product are the same when you install that product as a non-root user or a root user.
- Ensure that kernel parameters are updated as required. On Linux, kernel parameters are managed automatically for root installation but must be updated manually for non-root installations.
- On AIX, Asynchronous I/O (AIO) must be enabled. It is recommended the system has I/O Completion Ports (IOCP) enabled.
- Your home directory must be a valid DB2 path.

  DB2 installation paths have the following rules:
  - Can include lowercase letters (a-z), uppercase letters (A-Z), and the underscore character (_)

- – Cannot exceed 128 characters
- – Cannot contain spaces
- – Cannot contain characters from languages other than English
- – The path name cannot be a subdirectory of an existing DB2 installation.

About this task

There is nothing special a non-root user needs to do to install a DB2 database product, other than being logged in as a non-root user.

Procedure

To install your product as a non-root user:

1. Log in as a non-root user
2. Install your DB2 database product by using any of the methods available to you, including the following options:
   - The DB2 Setup wizard (GUI installation)
   - The **`db2setup`** command with a response file (silent installation)

     Note: Since non-root users cannot choose the directory where DB2 database products are installed, any **`FILE`** keyword in your response file is ignored.
3. After the DB2 database product is installed, you must open a new login session to use the non-root DB2 instance. Alternatively, you can use the same login session if you set up the DB2 instance environment with `*$HOME*/sqllib/db2profile` (for Bourne shell and Korn shell users) or `*$HOME*/sqllib/db2chsrc` (for C shell users), where *$HOME* is the non-root user's home directory.

What to do next

After the DB2 database product is installed, verify your operating system user process resource limits (**`ULIMITS`**). If the minimum **`ULIMITS`** values are not met, the DB2 engine can encounter unexpected operating resource shortage errors. These errors can lead to a DB2 database system outage.

Enabling root-based features in non-root installations with db2rfe

There are several features and abilities in non-root installations that are initially unavailable but can be enabled by running the **`db2rfe`** command.

Before you begin

This task requires root user authority.

Procedure

To enable the features and abilities that are initially unavailable in non-root installations:

1. Locate the sample configuration files. Two sample configuration files are provided:
   - `$HOME/sqllib/instance/db2rfe.cfg` is pre-configured with default values for the non-root DB2 instance
   - `$HOME/sqllib/cfg/db2rfe.cfg.sample` is not configured

   where `$HOME` is the non-root user's home directory.

2. Copy one of the sample configuration files to a different location so the original file remains unaltered.
3. Update the copied configuration file as needed. This configuration file is input to the **db2rfe** command.

   Here is a sample configuration file:

   ```
   INSTANCENAME=db2inst2
   SET_ULIMIT=NO
   ENABLE_HA=NO
   ENABLE_OS_AUTHENTICATION=NO
   RESERVE_REMOTE_CONNECTION=NO
      **SVCENAME=db2c_db2inst2
     **SVCEPORT=48000
   RESERVE_TEXT_SEARCH_CONNECTION=NO
     **SVCENAME_TEXT_SEARCH=db2j_db2inst2
     **SVCEPORT_TEXT_SEARCH=55000
   ```

   Note:

   - The value for the **INSTANCENAME** parameter is provided automatically by DB2 installer
   - The **SET_ULIMIT** parameter is available only on AIX. On other operating systems, a user with root authority needs to set **ULIMIT** values manually.
   - The default value for the other keywords is NO
   - Child parameters (such as **SVCENAME**) are commented out by default. Comments are denoted with **
   - If you set a parameter to YES, and if it has any child parameters, uncomment the child parameters and provide appropriate values. Any port values that are provided are examples. Ensure the port values that you assign are free.

   The following example is provided to show an edited configuration file that enables the features and abilities that are mentioned in the following list:

   - High availability
   - Operating system-based authentication
   - DB2 Text Search, with a service name of db2j_db2inst2 and a port value of 55000

   To enable these features and abilities, edit the configuration file as follows:

   ```
   INSTANCENAME=db2inst2
   SET_ULIMIT=NO
   ENABLE_HA=YES
   ENABLE_OS_AUTHENTICATION=YES
   RESERVE_REMOTE_CONNECTION=NO
      **SVCENAME=db2c_db2inst2
     **SVCEPORT=48000
   RESERVE_TEXT_SEARCH_CONNECTION=YES
       SVCENAME_TEXT_SEARCH=db2j_db2inst2
       SVCEPORT_TEXT_SEARCH=55000
   ```
4. Log in with root user authority.
5. Go to the $HOME/sqllib/instance directory, where $HOME represents the non-root user's home directory.
6. Run the **db2rfe** command by using the following syntax:

   ```
   db2rfe -f config_file
   ```

   where *config_file* is the configuration file that was created in Step 3.

What to do next

To keep root-based features enabled on non-root installations, rerun the **db2rfe** command after you apply fix packs or upgrade to a new version.

Applying fix packs to a non-root installation

The task of applying fix packs to a non-root installation is essentially the same as applying fix packs to a root installation, with a few exceptions.

Before you begin

Before you apply fix packs to a non-root installation, you must log on with the user ID that was used to install the non-root installation.

If you enabled root features in your non-root installation by using the **db2rfe** command, locate the configuration file that was used you run the **db2rfe** command. That configuration file will be needed to re-enable the root features after you apply the fix pack.

Procedure

To apply a fix pack to a non-root installation:

1. Apply your fix pack according to the "Applying fix packs" topic.

   Note: The **installFixPack** command's **-b** option is invalid for non-root installations.
2. Optional: Run the **db2rfe** command. If you had root-based features that were enabled in your non-root installation, and if you want to re-enable those features, the **db2rfe** command must be rerun. Running this command requires root user authority.

   Note: The `$HOME/sqllib/instance/db2rfe.cfg` configuration file does not get overwritten when you apply the fix pack. If you edited that configuration file when you first enabled root features, you can reuse that file when you run the **db2rfe** command. Check the `$HOME/sqllib/cfg/db2rfe.cfg.sample` file; if the fix pack introduced any new root features available to non-root installations, that file shows the new features.

Removing non-root DB2 database products by using db2_deinstall (Linux and UNIX)

This task provides steps for removing non-root DB2 database products or components by using the **db2_deinstall** command.

Before you begin

You must stop the non-root instance before you run the **db2_deinstall** command.

About this task

- This task applies to DB2 database products that were installed without root user authority. A separate task exists for uninstalling DB2 database products that were installed with root user authority.

- As with root users, non-root users can use the **db2_deinstall** command to uninstall DB2 database products. The **db2_deinstall** command for non-root installations has the same options as root installations, and has an extra option: **-f** sqllib.
- It is important to note that running **db2_deinstall** as a non-root user uninstalls the DB2 database product *and* drops the non-root instance. This outcome is different from root installations, where running **db2_deinstall** only uninstalls the DB2 database program files.
- You cannot remove DB2 database products by using a local operating system utility, such as **rpm** or SMIT.

Procedure

To uninstall a DB2 database product that was installed by a non-root user:

1. Log in with the user ID that was used to install the DB2 database product.
2. Go to the $HOME/sqllib/install directory, where $HOME is your home directory.
3. Run the **db2_deinstall** command.

   Note:

   - If you run the **db2_deinstall** command with the **-a** option, the DB2 database program files are removed, but any configuration files are left behind in a backup directory called sqllib_bk.
   - If you run the **db2_deinstall** command with the **-a -f sqllib** option, the entire sqllib subdirectory in your home directory is removed. If you have any files in sqllib you want to keep, be sure to copy them elsewhere before you run **db2_deinstall -a -f sqllib**.
   - As with root installations, running the **db2_deinstall** command with the **-F** option against a non-root installation allows the non-root user to remove specific DB2 features.

Index