



Geoportal Server 1.2.2 Installation Guide For Linux

Contents

| | |
|--|----------|
| 1. INTRODUCTION | 1 |
| 2. SET UP THE DATABASE | 1 |
| 2.1. ORACLE | 1 |
| 2.2. POSTGRESQL | 3 |
| 2.3. MYSQL | 6 |
| 3. DEPLOY AND CONFIGURE THE GEOPORTAL APPLICATION | 8 |

1. INTRODUCTION

To install the geoportal on a Linux environment, first follow the first three sections of the Geoportal Server 1.2.2 Installation Guide. When you get to Section 4 – where the geoportal database schema is created – you will need to detour from the Geoportal Server 1.2.2 Installation Guide and follow the steps below.

2. SET UP THE DATABASE

Depending on the database software you use, the database setup instructions vary significantly. Oracle users should follow the steps in section 2.1, PostgreSQL users should skip to section 2.2, and MySQL users should skip to section 2.3.

CAUTION: For all database software – these instructions include running database scripts. If you run the database scripts on top of an existing geoportal installation, you will overwrite your existing geoportal database.

2.1. ORACLE

In this section you will set up the tablespace and schema that will be used for the geoportal. The geoportal should run in its own tablespace and schema. Creating the geoportal database schema involves running two scripts:

- A “grants” script that sets the user permissions for creating the geoportal schema.
- A “create schema” script that creates the table structure, procedures and triggers, and populates tables in the geoportal schema.

Follow the steps below to prepare the copy the Oracle database files, prepare the tablespace, and run the scripts.

2.1.1. *Copy the Oracle database files*

You will need to move the following files from the <Geoportal Dir>\Database Scripts\Oracle directory to your Linux machine:

- grants_linuxoracle.sh
- grants_oracle.sql
- create_schema_linuxoracle.sh
- schema_oracle.sql

2.1.2. *Setup the geoportal tablespace*

- Open the terminal window.

- Type: sqlplus /nolog



Tip: In the following commands, be sure to include the semicolons.

- SQL>connect sys/sys as sysdba;
- SQL>create tablespace geoportal datafile '<oracle installation directory>/oradata/geoportal.dbf' size 500M AUTOEXTEND ON;
- SQL>create user geoportal identified by geoportalpwd default tablespace geoportal temporary tablespace temp;
- SQL>quit.

2.1.3. *Run grants_linuxoracle.sh*

- Open a terminal window.
- Switch User to your Oracle user that can connect to the Database, e.g. su – oracle
 - IMPORTANT:** make sure that the oracle user has permissions to run the shell files.
- Change directories to point to where you have placed your database scripts
E.g./Database Scripts/Oracle.
- Run the grants_linuxoracle.sh file from the terminal window using the following parameters:
Usage: ./ grants_linuxoracle.sh [sys username] [sys password] [geoportal username]
Where
[sys username] is the username of the sys user in Oracle
[sys password] is the password of the sys user in Oracle
[geoportal username] is the geoportal user you are creating

Sample Input: ./grants_linuxoracle.sh sys sys geoportal

When the script finishes executing you will be returned back to the command prompt and a text file (grants.txt) will open. Check the grants.txt file for error messages. Do not continue with the next script until error messages are resolved.

2.1.4. *Run create_schema_linuxoracle.sh*

- Open a terminal window.
- Change directories to point to where you have placed your database scripts

E.g./Database Scripts/Oracle.

- Run the create_schema_linuxoracle.sh file from the terminal window using the following parameters:

Usage: ./create_schema_linuxoracle.sh [geoportal username] [geoportal password]


Where

[geoportal username] is the username of the geoportal schema owner.

[geoportal password] is the password of the geoportal schema owner.

Sample Input: ./create_schema_linuxoracle.sh geoportal geoportalpwd

- When the script finishes executing you will be returned back to the command prompt and a text file (GPT_Schema.txt) will open. Check the GPT_Schema.txt file for error messages. Error messages and warnings that state a table or view does not exist can be ignored - It simply means that the script was trying to delete a nonexistent table.
- If there are no problematic errors in the GPT_Schema.txt file, then open the geoportal database in a database browser, and verify that the database and tables were created.

| Select | Schema  | Table Name | Tablespace |
|----------------------------------|--|---|------------|
| <input checked="" type="radio"/> | GEOPORTAL | GPT_HARVESTING_HISTORY | GEOPORTAL |
| <input type="radio"/> | GEOPORTAL | GPT_HARVESTING_JOBS_COMPLETED | GEOPORTAL |
| <input type="radio"/> | GEOPORTAL | GPT_HARVESTING_JOBS_PENDING | GEOPORTAL |
| <input type="radio"/> | GEOPORTAL | GPT_RESOURCE | GEOPORTAL |
| <input type="radio"/> | GEOPORTAL | GPT_RESOURCE_DATA | GEOPORTAL |
| <input type="radio"/> | GEOPORTAL | GPT_SEARCH | GEOPORTAL |
| <input type="radio"/> | GEOPORTAL | GPT_USER | GEOPORTAL |

2.2. POSTGRESQL

Setting up a PostgreSQL database for the geoportal consists of copying the PostgreSQL database files, setting up database permissions, and creating the database schema. For the permissions and creating the database schema, two scripts are used:

- A “grants” script that sets the user permissions for creating the geoportal schema.
- A “create schema” script that creates the table structure, procedures and triggers and populates tables in the geoportal schema.

2.2.1. Copy the PostgreSQL database files

You will need to move the following files from <Geoportal Dir>\Database Scripts\PostgreSQL to your Linux machine:

- grants_linuxpg.sh
- grants_pg.sql
- create_schema_linuxpg.sh
- schema_pg.sql

2.2.2. Run grants_linuxpg.sh

- Open a terminal window.
- Switch User to your PostgreSQL user that can connect to the Database, e.g. su – postgres
 - **IMPORTANT:** make sure that the postgres user has permissions to run the shell files.
- Change directories to point to where you have placed your database scripts
E.g./Database Scripts/PostgreSQL.
- Run the grants_linuxpg.sh file from the terminal window using the following parameters:
Usage: ./grants_linuxpg.sh [host] [port] [database] [geoportal schema] [postgresUser] [geoportal User]
Where
[host] is the machine name hosting PostgreSQL
[port] is the port number of PostgreSQL. Default = 5432
[database] is the database name. Default = postgres
[geoportal schema] is the name for the geoportal schema. Default = geoportal
[postgresUser] is the user name to connect as (not the user name to create)
[geoportal User] is the name for the geoportal schema owner. Default = geoportal

Sample Input: ./grants_linuxpg.sh localhost 5432 postgres geoportal postgres
geoportal
- When prompted with the message “Enter password for new role:”, input the password for the geoportal user
- When prompted with the message “Enter it again:”, input the password for the geoportal user again.

When the script finishes executing you will be returned back to the command prompt and a text file (grants.txt) will open. Check the grants.txt file for any possible error messages.

2.2.3. *Run create_schema_linuxpg.sh*

- Run the create_schema_linuxpg.sh file from the terminal window using the following parameters:

Usage: ./create_schema_linuxpg.sh [host] [port] [geoportal database] [geoportal user]

Where

[host] is the machine name hosting PostgreSQL

[port] is the port number of PostgreSQL. Default = 5432

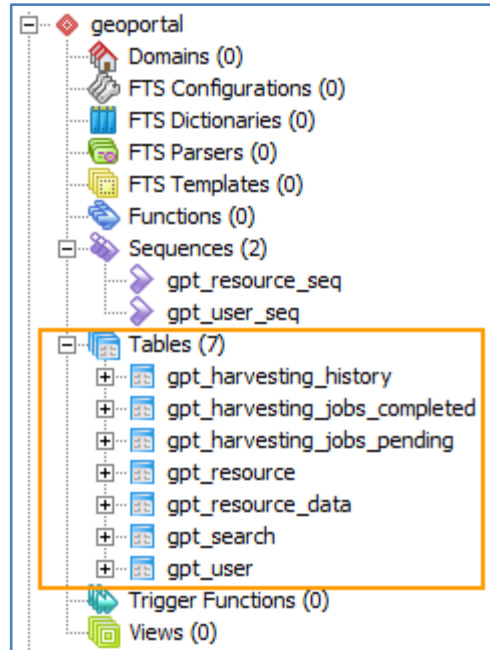
[geoportal database] is the database name that contains the geoportal schema.

Default = postgres

[geoportal user] is the name for the geoportal schema owner. Default = geoportal

Sample Input: ./create_schema_linuxpg.sh machineName 5432 postgres geoportal

- When prompted with the message “Enter password for geoportal user:”, input the password for the geoportal user
- When the script finishes executing you will be returned back to the command prompt and a text file (GPT_Schema.txt) will open. Check the GPT_Schema.txt file for error messages. Error messages and warnings that state a Table or view does not exist can be ignored. It simply means that the script was trying to delete a nonexistent table.
- Open the PostgreSQL Administrator tool.
- Verify that a new schema and tables were created.



2.3. MySQL

Setting up a MySQL database for the geoportal consists copying the database files, setting database permissions, and creating the database schema. Setting permissions and creating the schema is accomplished by running two scripts:

- A “grants” scripts that sets the user permissions for creating the geoportal schema
- A “create schema” script that creates the table structure, procedures, and triggers and populates tables in the geoportal schema.

It is assumed that you already have MySQL database software installed before running the scripts.

2.3.1. Copy the MySQL database files

You will need to move the following files from <Geoportal Dir>\Database Scripts\MySQL to your Linux machine:

- grants_mysql.sh
- create_schema_mysql.sh
- schema_mysql.sql

2.3.2. Run grants_mysql.sh

- Open a new terminal window.

- Change directories to point to where you have placed your database scripts, E.g./Database Scripts/MySQL.
- Run the grants_mysql.sh file from the terminal window using the following parameters:

Usage : ./grants_mysql.sh [dbserver] [port] [Geoportal database] [sys username] [sys password] [geoportal username] [geoportal server] [geoportal password]

Where [dbserver] is the machine name hosting MySQL

[port] is the port number of MySQL

[Geoportal database] is the database that contains the Geoportal Schema

[sys username] is the username of the sys user in MySQL

[sys password] is the password of the sys user in MySQL

[geoportal username] is the geoportal user

[geoportal server] is the name of the geoportal web application server

[geoportal password] is the geoportal user password

Sample Input: ./grants_mysql.sh localhost 3306 geoportal sys sys geoportal
geoportalserver geoportalpwd

- When the script finishes executing you will be returned back to the command prompt and a text file (grants_mysql.txt) will open. Check the grants_mysql.txt file for any possible error messages.



IMPORTANT: You must fix errors appearing in the grants_mysql.txt file; do not continue until the script runs without errors. Note that if you rerun the script after the user was successfully created, you may receive "ERROR 1396 (HY000): Operation CREAT USER failed..." You can ignore this error - see MySQL documentation for explanation
<http://bugs.mysql.com/bug.php?id=28331>.

2.3.3. Run create_schema_mysql.sh

- Run the create_schema_mysql.sh file from the command prompt window using the following parameters:

Usage : ./create_schema_mysql.sh [host] [port] [Geoportal database] [geoportal user]
[geoportal password]

Where [host] is the machine name hosting MySQL database

[port] is the port number of MySQL

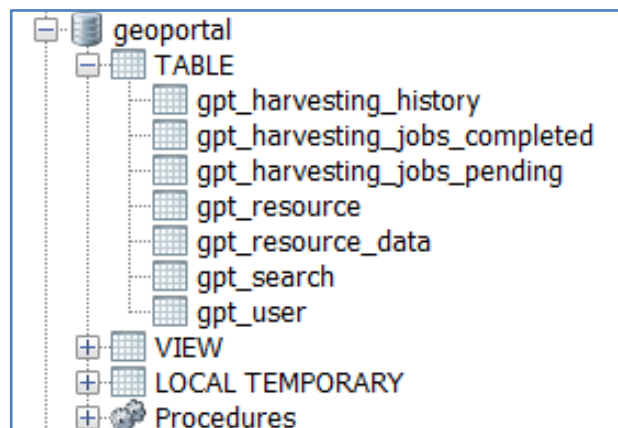
[Geoportal database] is the database that contains the Geoportal Schema

[geoportal user] is the geoportal user.

[geoportal password] is the geoportal password.

Sample Input: `./create_schema_mysql.sh localhost 3306 geoportal geoportal
geoportalpwd`

- When the script finishes executing you will be returned back to the command prompt and a text file (Geoportal_Schema.txt) will open. Check the Geoportal_Schema.txt file for any error messages. Error messages and warnings that state a table or view does not exist can be ignored. It simply means that the script was trying to delete a nonexistent table.
- Verify that a new database and tables were created.



3. DEPLOY AND CONFIGURE THE GEOPORTAL APPLICATION

After running the database scripts, the Geoportal Server 1.2 Installation Guide proceeds to instructions for deploying the geoportal. You will refer back to the Geoportal Server 1.2 Installation Guide, starting at Section 5 to complete the geoportal installation, **but first read the important note below.**

IMPORTANT: When configuring the geoportal web application – you will be modifying the gpt.xml file - be aware of one setting that will differ for Linux systems. The setting will need to be added in, and is for “httpClient.alwaysClose”. You will need to add this parameter to the gpt.xml file and set it to true. This needs to be done to avoid errors during harvesting. The cause is that Apache HTTP client doesn’t close socket connections by default if not explicitly told to do so by the server; in Linux each socket connection has its own file descriptor and imposes a limitation to the number of concurrently open files (invoke: `ulimit -n` to check that limitation; typically: 1024). This number can be quickly exhausted. To address this, the “httpClient.alwaysClose” parameter forces the Apache HTTP client to always close unused socket connections and seems to resolve the problem.

To add in the parameter to your geoportal configuration, follow instructions below.

- Follow instructions in Geoportal Server 1.2 Installation Guide section 5 to deploy your geoportal web application.
- Open the \\geoportal\WEB-INF\classes\gpt\config\gpt.xml file, and find the closing catalog parameter, </catalog>.
- Just before the </catalog> tag, add the following:

```
<parameter key="httpClient.alwaysClose" value="true"/>
```

- Return to the Geoportal Server 1.2 Installation Guide to continue with configuring the gpt.xml file.