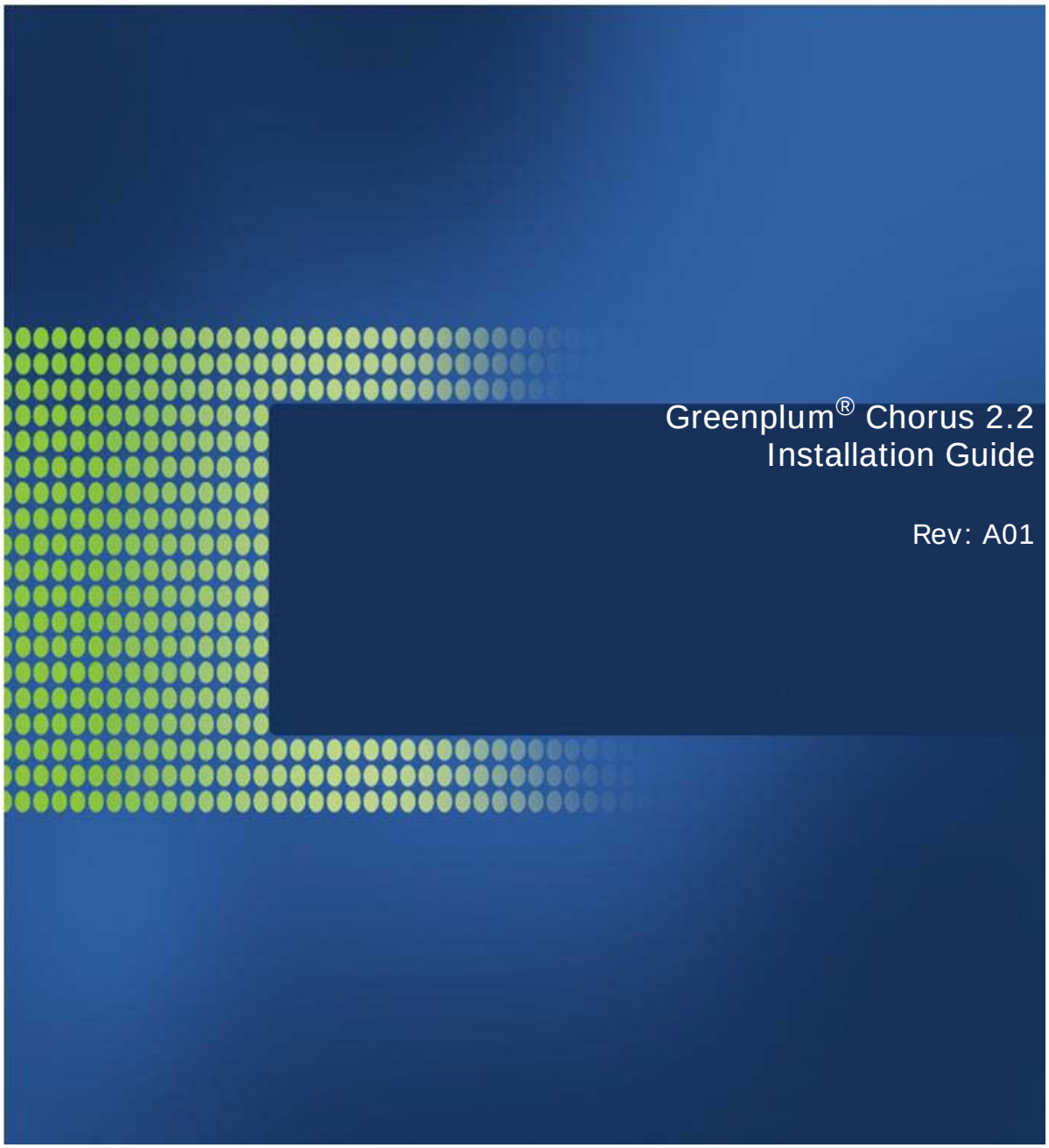

Greenplum® Chorus 2.2 Installation Guide

Rev: A01

December 10, 2012

Greenplum Chorus Installation Guide 2.2 - Contents

Preface

This guide describes the tasks you must do to install and start the Greenplum Chorus system.

About This Guide

This guide provides information and instructions for installing and initializing a Greenplum Chorus system. This guide is intended for system administrators responsible for building a Greenplum Chorus system.

This guide assumes knowledge of Linux/Unix system administration, database management systems, database administration, and structured query language (SQL).

This guide contains the following chapters:

- Chapter 1, “Introduction to Greenplum Chorus”— Information about Greenplum Chorus.
- Chapter 2, “Installing or Upgrading Greenplum Chorus”— Guidelines for installing a Greenplum Chorus system.
- Chapter 3, “Configuring Greenplum Chorus 2.2” — Guidelines for configuring a Greenplum Chorus system.

Document Conventions

The following conventions are used throughout the Greenplum Chorus documentation to help you identify certain types of information.

Text Conventions

Table 0.1 Text Conventions

Text Convention	Usage	Examples
bold	Button, menu, tab, page, and field names in GUI applications	Click **Cancel** to exit the page without saving your changes.
italics	New terms where they are defined Database objects, such as schema, table, or columns names	The *master instance* is the `postgres` process that accepts client connections. Catalog information for Greenplum Chorus resides in the *pg_catalog* schema.
`monospace`	File names and path names Programs and executables Command names and syntax Parameter names	Edit the `postgresql.conf` file. Use `gpstart` to start Greenplum Chorus.
`<monospace italics>`	Variable information within file paths and file names Variable information within command syntax	`/home/gpadmin/<config_file>` `COPY <tablename> FROM 'filename'`
`monospace bold`	Used to call attention to a particular part of a command, parameter, or code snippet.	Change the host name, port, and database name in the JDBC connection URL: `jdbc:postgresql://host:5432/mydb`
`UPPERCASE`	Environment variables SQL commands Keyboard keys	Make sure that the Java `/bin` directory is in your `$PATH`. `SELECT * FROM my_table;` Press `CTRL+C` to escape.

Command Syntax Conventions

Table 0.2 Command Syntax Conventions

Text Convention	Usage	Examples
`{ }`	Within command syntax, curly braces group related command options. Do not type the curly braces.	`FROM { '<filename>' | STDIN }`
`[ ]`	Within command syntax, square brackets denote optional arguments. Do not type the brackets.	`TRUNCATE [ TABLE ] <name>`
`...`	Within command syntax, an ellipsis denotes repetition of a command, variable, or option. Do not type the ellipsis.	`DROP TABLE <name> [, ...]`
`|`	Within command syntax, the pipe symbol denotes an "OR" relationship. Do not type the pipe symbol.	`VACUUM [ FULL | FREEZE ]`
`$ system_command` `# root_system_command`	Denotes a command prompt - do not type the prompt symbol. $ and # denote terminal command prompts.	`$ createdb mydatabase` `# chown gpadmin -R /datadir`

Getting Support

EMC support, product, and licensing information can be obtained as follows.

Product information

For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Powerlink Web site (registration required), and choose the EMC Download Center.

Technical support

For technical support, go to Powerlink and choose **Support**. On the Support page, you will see several options, including one for making a service request. Note that to open a service request, you must have a valid support agreement. Please contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

1. Introduction to Greenplum Chorus

Greenplum Chorus is a collaborative platform for data science. Chorus users iterate faster and finish projects sooner through secure access to data and by sharing content and findings within their organization through a platform especially built for this purpose.

Organizations will value the empowerment of data science, along with the reduction of IT operational involvement and one-off infrastructure costs.

This chapter focuses on how you can prepare your environment for Greenplum Chorus. In particular, this chapter describes the following topics:

- System Requirements
- Where to go from here

System Requirements

This section describes the system requirements for Greenplum Chorus 2.2.

You can install Chorus 2.2:

- on the Standby Master of a Greenplum DCA.
- on any Linux server with an Intel Pentium Pro compatible (P3/Athlon and above) CPU and 8GB of RAM.

In either case, Greenplum recommends 500GB of free disk space.

Prerequisites

1. Check (with, for example, `cat/etc/redhat-release`) that your system meets one of the following operating system requirements:
   - Red Hat Enterprise Linux 5.5, 5.7, 6.2 (64 bit)
   - CentOS 5.5, 5.7, 6.2 (64 bit)
   - SuSE Linux Enterprise Server 11 (64 bit)
   - OSX Lion x86_64
2. Verify (with `java -version`) that you have JRE 1.6.0_21 or later installed. JRE 1.7 is not supported.

   Note: If you are installing on a DCA, a correct version of Java will have been pre-installed under `/usr/java/latest`. Greenplum recommends that you use this same path for a non-DCA installation.
3. Check that the variable, `JAVA_HOME`, is set correctly (see “Add the following line to your .bashrc file to set the JAVA_HOME variable:” on page 10).
4. Verify that you have one of the following supported browsers:
   - Firefox 14.0 or later
   - Google Chrome 20 or later.
   - Internet Explorer 8.0 with Google Chrome Frame
   - Internet Explorer 9.0 (Google Chrome Frame not required)

   Note: IE 9 can be made to simulate IE 7 or IE 8 in its “compatibility mode.” Chorus does not work with IE 7 or 8 (without Chrome frame), so you must disable compatibility mode. To do this:
   a. Press the Alt key to open the IE9 menu bar.
   b. Choose the **Tools** menu.
   c. If Compatibility View is unchecked, do nothing.
   d. If Compatibility View is checked, select it to uncheck it.

5. Edit the `pg_hba.conf` file for any Greenplum Database instances that will be connected to Chorus so that the GPDB instance will accept connections from all users of the Chorus server. See the *Greenplum Database Administrator Guide* for how to do this.

6. If you want to use Chorus to facilitate creating external tables that point to Hadoop, you merely need to configure Hadoop to work with the Greenplum Database (GPDB). See the *Greenplum Database Administrator Guide* for instructions on setting up Hadoop with GPDB. Chorus does not require an installation of Hadoop, however.

Where to go from here

Proceed to "Installing or Upgrading Greenplum Chorus" on page 9 for information on how to install or upgrade to Greenplum Chorus 2.2.

2. Installing or Upgrading Greenplum Chorus

This chapter describes how you can install Greenplum Chorus or upgrade from a previous version of the product. The following topics are included:

Note: Greenplum Chorus 2.2, does not support an online upgrade, so you will need to stop and restart Chorus even if you apply a patch.

Preparing to Install Greenplum Chorus 2.2

1. Make sure that you have met the requirements listed in "System Requirements" on page 6.

2. If you are installing on a DCA, use PuTTY to establish an `ssh` connection to the GPDB Standby Master.

3. Create the user, `chorus`, at the shell prompt:

```
# useradd chorus
# groupadd chorus
# passwd chorus
```

Note: When you enter the command `passwd chorus` you are asked for the password: for a DCA installation, choose `chorus`; for a non-DCA installation, you can choose anything you like.

Choosing `chorus` as the password will bring up a message (as in the example below) about not using a dictionary word. `chorus` will be accepted, however, after you enter it a second time. Here is an example:

```
[root@smdw /]# passwd chorus
Changing password for user chorus.
New UNIX password:<Enter chorus here>
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:<Enter chorus again>
passwd: all authentication tokens updated successfully.
[root@smdw /]#
```

Although it doesn't tell you, `chorus` has been accepted as the password.

4. Switch to user `chorus`:

```
# su - chorus
```

5. Update your `.bashrc` file:
   1. Open your text editor:
      ```
      $ vi ~/.bashrc
      ```
   2. Add the following line to your `.bashrc` file to set the `JAVA_HOME` variable:
      ```
      export JAVA_HOME=/usr/java/latest
      ```
   3. Close the `.bashrc` file and source it to activate the changes:
      ```
      $ source ~/.bashrc
      ```
6. Create installation and data directories (as `root` user):
   1. Create the path for the installation binaries:
      ```
      # mkdir -p /usr/local/greenplum-chorus
      # chown -R chorus:chorus /usr/local/greenplum-chorus
      ```
   2. Create the path for shared data:
      ```
      # mkdir -p /data/greenplum-chorus
      # chown -R chorus:chorus /data/greenplum-chorus
      ```
      `/usr/local/greenplum-chorus` and `/data/greenplum-chorus` are the directories that will be suggested to you when running the installation script. You can substitute any directories of your choice as long as they are owned by the chorus user.

   Note: The location for shared data should have >= 500GB of available space. You can run the `df -h` command as `root` to see the free space you have on your mounted file systems.

Important: If you are installing on a DCA, skip steps 7 through 9.

7. Set the following parameters in `/etc/security/limits.conf`:
   ```
   soft nofile 65536
   hard nofile 65536
   soft nproc 131072
   hard nproc 131072
   ```
8. Set the following parameters in `/etc/sysctl.conf`:
   ```
   kernel.shmmax = 500000000
   kernel.shmall = 4000000000
   ```
9. Restart the server if you made changes to the configuration parameters in steps 7 and 8.

Installing Greenplum Chorus 2.2

1. Go to the EMC Download Center to download the Greenplum Chorus installation package. The package will be in the form
   `greenplum-chorus-2.2.0.0.<build number>-<sha>.sh`
   where `sha` is a hash that maps to a specific code commit.

2. Save the package to a folder where the `chorus` user has write privileges. In the case of a DCA installation, this folder should be on the DCA Standby Master in the `/home/chorus` directory.

3. Run MD5 on the binary. This generates a string which you can compare to the value listed on Powerlink in order to verify that you have downloaded the correct file. For example, running
   ```
   # md5sum ~/greenplum-chorus-2.2.0.0.733-66d63951e.sh
   ```
   on the DCA smdw might return
   ```
   MD5 greenplum-chorus-2.2.0.0.733-66d63951e.sh=
   bd00870bac943790fa032cc7a2651af
   ```
   You can compare `bd00870bac943790fa032cc7a2651a` with the value listed on Powerlink. If the values match, you have downloaded the correct file.

4. The installation package is a self-extracting script that contains the following components.
   - Chorus code
   - PostgreSQL database package

   Replace the `<version>`, `<build.number>`, and `<sha>` with the real version string and build number. Then run
   ```
   # chmod +x ~/greenplum-chorus-2.2.0.0.733-66d63951e.sh
   ```
   Running this command gives you binary execution privileges.

5. Log in as the `chorus` user and run the installer
   Replace the `<version>`, `<build.number>`, and `<sha>` with the real version string and build number. Then run this command to execute the installer:
   ```
   $ ./greenplum-chorus-2.2.0.0.733-66d63951e.sh
   ```

6. Provide the installer with the following information to continue the process:
   a. Type `y` to accept the license agreement; otherwise the installer will exit.
   b. Provide the correct directory if you want the installation binaries at a location different from `/usr/local/greenplum-chorus`
   c. Provide the correct directory if you want shared data at a location different from `/data/greenplum-chorus`

      Ensure that you have adequate disk space for shared data; Greenplum recommends a minimum of 500 GB. You can run the `df -h` command as `root` to see the free space you have on your mounted file systems.

7. The installer validates operating system compatibility and displays an error message listing the expected operating systems if your OS is not one of them. You can respond to the error message by choosing one of the listed OS that is equivalent to your OS.

8. You are prompted to enter your passphrase, which can be any combination of alphanumeric characters. This will be used to generate a secret key to be used for recovering passwords from the GPDB. Write it down and keep it in a safe place!

   Note: The secret key is kept (encrypted) in a file named `secret.key`, located under the `shared` data directory.

When finished, the installer exits.

9. Prior to starting the Chorus server, you should review the contents of the `chorus.properties` file. See Chapter 3, "Configuring Greenplum Chorus 2.2". Also, Greenplum recommends that Chorus is configured with an ssl certificate (see "To generate an SSL certificate with OpenSSL" on page 19).

10. To start the Chorus server, do the following as user `chorus`:

```
$ source /usr/local/greenplum-chorus/chorus_path.sh
$ chorus_control.sh start
```

To verify your Greenplum Chorus installation

Make sure your external network can use Greenplum Chorus with these steps.

1. From an external server, log into Greenplum Chorus:

   `http://<external IP address smdw>:8080` (DCA installation)

   `http://<external IP address>:8080` (non-DCA installation)

2. Log into Greenplum Chorus with the user name `chorusadmin` and the password `secret`

3. Ensure Greenplum Chorus loads in the browser.

Note: Greenplum recommends that you change your username and password after you have verified that your installation works.

Important: After installing Chorus and verifying the installation, you should set up a daily backup. See "Backing up Greenplum Chorus" on page 24.

Upgrading to Greenplum Chorus 2.2

This topic describes how you can upgrade from Greenplum Chorus 2.1.x.x to Greenplum Chorus 2.2.

Note: If you want to upgrade from Chorus 2.0.x.x to 2.2, you must first upgrade from 2.0 to 2.1 and then upgrade from 2.1 to 2.2.

This topic describes the following tasks:

- To prepare for your upgrade
- To run the upgrade

To prepare for your upgrade

1. Back up any previous installations of Greenplum Chorus 2.1 before you begin the upgrade process. Refer to the *Greenplum 2.1 Installation Guide* for 2.1 backup instructions.

2. Go to the EMC Download Center to download the Greenplum Chorus installation package. The package will be in the form
   `greenplum-chorus-2.2.0.0.<build number>-<sha>.sh`
   where `sha` is a hash that maps to a specific code commit.

3. Put the installation package in a folder where the user `chorus` has write privileges. In the case of a DCA installation, this folder should be on the DCA Standby Master in the `/home/chorus` directory.

4. Run MD5 on the binary. This generates a string which you can compare to the value listed on Powerlink in order to verify that you have downloaded the correct file. For example, running
   ```
   # md5sum ~/greenplum-chorus-2.2.0.0.733-66d63951e.sh
   ```
   on the DCA smdw might return
   ```
   MD5 greenplum-chorus-2.2.0.0.733-66d63951e.sh=
   bd00870bac943790fa032cc7a2651af
   ```
   You can compare `bd00870bac943790fa032cc7a2651a` with the value listed on Powerlink. If the values match, you have downloaded the correct file.

5. Log in as `chorus`.
   Since the software can only be upgraded by the user who has the privileges to start and stop the system, you must log in as `chorus`

6. Go to the existing `chorus` install directory and source `edc_path.sh`. For example:
   ```
   $ cd /data/chorus/
   $ source edc_path.sh
   ```

7. Create *new* installation and data directories (as `root` user). This step creates the directories as root user and gives ownership of the directories to the chorus user. These directories must have different names than the current directories. For example:

   1. Create the path for the installation binaries:
   ```
   # mkdir -p /usr/local/greenplum-chorus
   # chown -R chorus:chorus /usr/local/greenplum-chorus
   ```

   2. Create the path for shared data:
   ```
   # mkdir -p /data/greenplum-chorus
   # chown -R chorus:chorus /data/greenplum-chorus
   ```

   Note: The location for shared data should have >= 500GB of available space. You can run the `df -h` command as `root` to see the free space you have on your mounted file systems.

To run the upgrade

1. The installation package is a self-extracting script that contains the following components.
   - Chorus code
   - PostgreSQL database package

   Log in as `root` and replace the `<version>`, `<build.number>`, and `<sha>` with the real version string and build number. Then run:

   ```
   # chmod +x ~/greenplum-chorus-2.2.0.0.733-66d63951e.sh
   ```

   Running this command gives you binary execution privileges.

2. Log in as `chorus` and replace the `<version>`, `<build.number>`, and `<sha>` with the real version string and build number. Then run the installer with this command:

   ```
   $ ./greenplum-chorus-2.2.0.0.733-66d63951e.sh
   ```

3. The installer will ask for an installation directory. Enter the install directory for the 2.1 installation.

Important: You must enter the installation directory for the *existing* 2.1 installation. This causes the installer to verify that the installation is upgradable. If it is upgradable, you are asked whether to proceed. Answer yes.

4. Provide the installer with the following information to continue the process:
   a. Type `y` to accept the license agreement; otherwise the installer will exit.
   b. Provide the name and path of the new directory where you want to install the installation binaries. Preferably:

      ```
      /usr/local/greenplum-chorus
      ```

   c. Provide the name and path of the new directory where you want to install shared data. Preferably:

      ```
      /data/greenplum-chorus
      ```

      Ensure that you have adequate disk space for shared data; Greenplum recommends a minimum of 500 GB.

5. The installer validates operating system compatibility and displays an error message listing the expected operating systems if your OS is not one of them. You can respond to the error message by choosing one of the listed OS that is equivalent to your OS. If your OS is not an equivalent, the installer exits.

6. You are prompted to enter your passphrase, which can be any combination of alphanumeric characters. This will be used to generate a secret key to be used for recovering passwords from the GPDB. Write it down and keep it in a safe place!

   When finished, the installer exits.

7. Prior to starting the Chorus server, you should review the contents of the `chorus.properties` file. See Chapter 3, “Configuring Greenplum Chorus 2.2”. Also, Greenplum recommends that Chorus is configured with an ssl certificate (see “To generate an SSL certificate with OpenSSL” on page 19).

8. To start the Chorus server, do the following as user `chorus`:

```
$ source /usr/local/greenplum-chorus/chorus_path.sh
$ chorus_control.sh start
```

To verify your Greenplum Chorus installation

Make sure your external network can use Greenplum Chorus with these steps.

1. From an external server, log into Greenplum Chorus:

   `http://<external IP address smdw>:8080` (DCA installation)

   `http://<external IP address>:8080` (non-DCA installation)

2. Log in with an existing Chorus user.

3. Ensure Greenplum Chorus loads in the browser.

4. When you have verified the installation, remove the directories you used for Chorus 2.1.

Note: Greenplum recommends that you change your username and password after you have verified that your installation works.

Important: After installing Chorus and verifying the installation, you should set up a daily backup. See "Backing up Greenplum Chorus" on page 24.

Starting and Stopping Greenplum Chorus 2.2

1. Log in as user, `chorus`.

Important: You should not perform these tasks as `root`.

2. Run the commands to perform each of the following tasks.

To start Greenplum Chorus

```
$ cd <chorus install path>
$ source chorus_path.sh
$ chorus_control.sh start
```

To stop Greenplum Chorus

```
$ cd <chorus install path>
$ source chorus_path.sh
$ chorus_control.sh stop
```

To restart Greenplum Chorus

```
$ cd <chorus install path>
$ source chorus_path.sh
$ chorus_control.sh restart
```

To monitor Greenplum Chorus

Monitoring consists of checking that all chorus processes are running and restarting any processes that are down.

```
$ cd <chorus install path>
```

```
$ source chorus_path.sh
$ chorus_control.sh monitor
```

To backup Chorus data

```
$ cd <chorus install path>
$ source chorus_path.sh
$ chorus_control.sh backup [-d dir] [-r days]
```

where `-d` supplies the directory for the backup and `-r` specifies how many days of backup files should be kept in the backup directory. Files more than r days old will be removed.

Important: Greenplum recommends running a cron job to backup chorus at least daily. See "Backing up Greenplum Chorus" on page 24.

To start / stop / restart individual Greenplum Chorus services only

Chorus consists of five services: `postgres`, `workers`, `scheduler`, `solr`, and `webserver`. The `start`, `stop`, `restart`, and `monitor` commands apply to all services at once. For example, `chorus_control.sh start` starts all services.

You can also start, stop, restart, and monitor individual services, as follows:

```
chorus_control.sh start <service_name>
chorus_control.sh stop <service_name>
chorus_control.sh restart <service_name>
chorus_control.sh monitor <service_name>
```

where *service_name* is the name of one of the five individual services.

Where to go from here

If you have completed installing or upgrading Greenplum Chorus, proceed to "Configuring Greenplum Chorus 2.2" on page 17.

3. Configuring Greenplum Chorus 2.2

This chapter describes how you configure the specific properties in Greenplum Chorus. This chapter describes the following:

Setting up Greenplum Chorus 2.2

You may need to configure certain properties to run Greenplum Chorus. This topic includes descriptions for the following tasks:

To configure or change the HTTP port number

The default HTTP port for Greenplum Chorus is 8080. You can change it to any free port number above 1024.

1. Edit the `<installation directory>/shared/chorus.properties` file. Change the `server_port` entry to the port number you want. For example:

   ```
   server_port= 1550
   ```

2. Restart Greenplum Chorus.

Note: If `ssl` is enabled and configured, this HTTP port will redirect to the `ssl_server_port` (see "Generating and Installing the SSL Certificate" on page 19).

To configure or change the PostgreSQL Database port number

The default port number for the PostgreSQL database listening is 8543. You can change it to any free port number above 1024.

1. Edit the <*installation directory*>/shared/chorus.properties file. Change the postgres_port entry to the port number you want. For example:

   ```
   postgres_port= 9000
   ```

2. Restart Greenplum Chorus.

To configure or change the Solr port number

The default port number for Solr is 8983. You can change it to any free port number above 1024.

1. Edit the <*installation directory*>/shared/chorus.properties file. Change the solr_port entry to the port number you want For example:

   ```
   solr_port= 9001
   ```

2. Restart Greenplum Chorus.

To configure parameters for the Java Virtual Machine

1. Edit the <*installation directory*>/shared/chorus.properties file. Change the java_options entry as you wish. For example:

   ```
   java_options=-Djava.library.path=$CHORUS_HOME/vendor/hadoop/
   lib/ -server -Xmx1024m -Xms512m -XX:MaxPermSize=128m
   ```

2. Restart Greenplum Chorus.

To configure the indexing frequency of database instances

1. Edit the <*installation directory*>/shared/chorus.properties file. Change the reindex_datasets_interval_hours entry to the time interval you want. For example:

   ```
   reindex_datasets_interval_hours= 24
   ```

2. Restart Greenplum Chorus.

To configure an external server to import data with gpfdist

To enable data movement between databases, gpfdist must be installed and running on the Chorus host. Two processes must be started: Start one process for writing and one process for reading, each with different ports but pointing to the same directory.

See the *Greenplum Database Administrator Guide* on how to configure gpfdist.

1. Download the gpfdist package and install it.

2. Examine the gpfdist entry in <*installation directory*>/shared/chorus.properties. For example,

   ```
   gpfdist.ssl.enabled= false
   ```

   Note: Set gpfdist.ssl.enabled to true if gpfdist is configured with ssl certificates. ssl certificates must be installed on all segment servers.

   ```
   gpfdist.url= sample-gpfdist-server
   gpfdist.write_port= 8000
   gpfdist.read_port= 8001
   ```

```
gpfdist.data_dir= /tmp
```

3. Start `gpfdist` with the `write_port` value and the `data_dir` value.
4. Start `gpfdist` with the `read_port` value and the `data_dir` value.
5. Restart `chorus` to activate the changes.

To run data_import

For more complete information about `gpfdist`, go to Powerlink and refer to *The Greenplum Database Administrator Guide 4.2*.

Generating and Installing the SSL Certificate

Greenplum recommends that you configure Greenplum Chorus with an SSL certificate. There are several ways to do this, including setting up a web server in front of Chorus, or installing the certificate on the load balancer.

To generate an SSL certificate with OpenSSL

Note: If you are using a self-signed certificate, your browser will prompt you with an untrusted SSL certificate warning

1. Generate an RSA private key

   ```
   openssl genrsa -des3 -out server.key 1024
   ```

   Use anything for your password that you will remember later.

2. Generate a Certificate Signing Request (CSR)

   ```
   openssl req -new -key server.key -out server.csr
   ```

   Respond to the questions as shown in the examples:

   ```
   What is your first and last name?
   [Unknown]:  chorus-ga.greenplum.com
   Note: Enter the URL for Greenplum Chorus.
   What is the name of your organizational unit?
   [Unknown]:  Data and Insights
   What is the name of your organization?
   [Unknown]:  Greenplum
   What is the name of your City or Locality?
   [Unknown]:  San Mateo
   What is the name of your State or Province?
   [Unknown]:  California
   What is the two-letter country code for this unit?
   [Unknown]:  US
   Is CN=chorus-ga.greenplum.com, OU=Data and Insights,
   O=Greenplum, L=San Mateo, ST=California, C=US correct?
   [no]:  yes
   Enter key password for <chorus>
   ```

```
(RETURN if same as keystore password.)
```

3. Remove Passphrase from Key

   ```
   cp server.key server.key.org
   openssl rsa -in server.key.org -out server.key
   ```

   Without this step you will need to type the password you created in Step 1 each time you start Chorus.

4. Generate a self-signed certificate from the CSR

   Note: If you want an official SSL certificate (Greenplum recommended), submit this CSR to a signing authority such as Thawte or Verisign and continue to Step 5 when you have the certificate (.crt) file.

   ```
   openssl x509 -req -days 365 -in server.csr -signkey
                                server.key -out server.crt
   ```

5. Install the Private Key and Certificate into Chorus

   Configure `chorus.properties` to point to the locations of your private key and certificate files:

   ```
   ssl.enabled= true
   ssl_server_port= 8443
   ssl_certificate=
          /usr/local/greenplum-chorus/current/config/test.crt
   ssl_certificate_key=

   /usr/local/greenplum-chorus/current/config/test.key
   ```

Restart Chorus to apply the configuration.

Note: To run Chorus on port 443 (the default ssl port e.g. https://:443), Greenplum recommends that you set up a Web server proxy to Chorus.

Enabling LDAP Support

By default, Greenplum Chorus 2.2 manages users through the database. Greenplum Chorus uses the LDAPv3 server, including Active Directory support, to manage and authenticate users. For more information about the LDAP server, see http://www.ietf.org/rfc/rfc2251.txt.

Enabling LDAP provides the following benefits:

- Adding users to Greenplum Chorus: Once a user is added into Chorus, Chorus maintains a read-only copy of common user information, such as the user's name and department.
- Authenticating users with LDAP.

Configuring LDAP

1. Try connecting to your AD or LDAP installation with a separate LDAP exploration tool to ensure that all configuration properties are correct prior to attempting to configure these in Chorus.

2. Edit the <*installation directory*>/shared/chorus.properties file to configure LDAP in Chorus.

3. Change the default entries for the following properties, if desired:

```
ldap.host= 10.32.88.212
ldap.enable= false
ldap.port= 389
ldap.connect_timeout= 10000
ldap.bind_timeout= 10000
ldap.search.timeout= 20000
ldap.search.size_limit= 200
ldap.base= DC=greenplum,DC=com
ldap.user_dn= greenplum\chorus
ldap.password= secret
ldap.dn_template= greenplum\{0}
ldap.attribute.uid= sAMAccountName
ldap.attribute.ou= department
ldap.attribute.gn= givenName
ldap.attribute.sn= sn
ldap.attribute.cn= cn
ldap.attribute.mail= mail
ldap.attribute.title= title
```

4. Restart the server to complete certificate configuration.

The following table contains a list and description of properties related to LDAP:

Table 3.1 LDAP configuration parameters

LDAP Parameters	Description
ldap.enable	boolean value to enable or disable ldap. (false by default).
ldap.host	LDAP server IP or host name.
ldap.port	LDAP server port.
ldap.search.size_limit	LDAP search match number limitation. (100 by default)
ldap.base	LDAP base DN.
ldap.user-dn	LDAP credential used to search against LDAP server. If LDAP server support anonymous search, this could be commented out.
ldap.password	This password corresponds to the chorus.ldap.userDN field. If LDAP server supports anonymous search, this field can be commented out.
ldap.dn_template	DN template
ldap.attribute.uid	This is a required field. For Active Directory, this is often sAMAccountName. This is the LDAP username attribute (“uid'' by default)

Table 3.1 LDAP configuration parameters

LDAP Parameters	Description
ldap.attribute.ou	LDAP attribute name for Organizational Unit or Department (“ou'' by default)
ldap.attribute.gn	LDAP attribute name for First name (“gn'' by default)
ldap.attribute.sn	LDAP attribute name for Last name. (“sn'' by default)
ldap.attribute.mail	LDAP attribute name for e-mail address. (“mail'' by default)
ldap.attribute.title	LDAP attribute name for User’s title. (“title'' by default)

Customizing chorus.properties

The following table lists and describes other relevant chorus.properties:

Table 3.2 The chorus.properties file

Parameter	Description
`session_timeout_minutes= 480`	Expiration of the access ticket in minutes. Default is 480 (8 hours)
`instance_poll_interval_minutes= 5`	Interval at which the system polls to see that instances are online. Uses instance owner’s credentials for polling.
`delete_unimported_csv_files_interval_hours= 1`	Interval for deleting files on which work has been abandoned.
`delete_unimported_csv_files_after_hours= 1`	Time after which a csv file uploaded to Chorus server for import will be deleted, if import has not yet been initiated.
`reindex_search_data_interval_hours= 24`	Interval for recrawling the instances.
`sandbox_recommended_size_in_gb= 5`	Sandbox related setting, default unit is GB. **Note:** This value provides a visual indicator that indicates when a workspace's sandbox exceeds the recommended size.

Table 3.2 The chorus.properties file

Parameter	Description
`worker_threads= 1` `webserver_threads= 20`	Configuring the thread pool size of webserver and worker processes: The # of webserver threads determines the maximum number of simultaneous web requests. The # of worker threads determines the maximum number of asynchronous jobs, such as table copying or importing, that can be run simultaneously. Each web or worker thread may use its own connection to the local Postgresql database. Therefore, the sum of 'worker_threads' + 'webserver_threads' must be less than the 'max_connections' configured in `postgresql.conf`. The 'max_connections' parameter may be based on your operating system's kernel shared memory size. For example, on OS X this parameter will default to 20.
`file_sizes_mb.workfiles= 10`	Maximum upload work file size.
`file_sizes_mb.csv_imports= 100`	Maximum size for imported files.
`file_sizes_mb.user_icon= 5`	Maximum size for the user icon.
`file_sizes_mb.workspace_icon= 5`	Maximum size for the workspace icon.
`file_sizes_mb.attachment= 10`	Maximum size for file attachments.
`logging.syslog.enabled= false`	If true, logs are written to syslog rather than to files.
`tableau.enabled= true`	If false, tableau is disabled even if other tableau parameters are specified.
`tableau.url= <ip address>`	URL of tableau server.
`tableau.port= 80`	Tableau server port.
`gnip.enabled= true`	Enables gnip account.
`gnip.csv_import_max_file_size_mb= 50`	Maximum size of chunks of gnip data downloaded.
`kaggle.enabled= true`	If false, kaggle is disabled even if other tableau parameters are specified.
`kaggle.api_key=<key provided on request>`	Key to access kaggle.
`default_preview_row_limit = 500`	Maximum preview rows.
`execution_timeout_in_minutes = 300`	Workfile execution timeout in minutes.

Backing up Greenplum Chorus

Make sure that Greenplum Chorus is up when you back up the database. During the backup process, the following backup file is dumped to your backup directory:

```
greenplum_chorus_backup_YYYYMMDD_HHMMSS.tar
```

where YYYYMMDD_HHMMSS is a timestamp.

Here is the procedure:

```
$ cd <chorus install path>
$ source chorus_path.sh
$ chorus_control.sh backup [-d dir] [-r days]
```

`-d` supplies the directory for the backup. If you do not specify a backup directory, the backup utility creates the default backup directory

```
/data/greenplum-chorus/bak
```

`-r` specifies how many days of backup files should be kept in the backup directory. Files more than `r` days old will be removed. If `r` is not specified, no files are removed.

For example, the following command backs up the Greenplum Chorus files to `data/greenplum-chorus/daily_bu` and deletes backup files that are more than 10 days old.

```
chorus_control.sh backup -d /data/greenplum-chorus/daily_bu
                                                        -r 10
```

Note: Greenplum Chorus logs and indexes are not stored in the backup file. Greenplum recommends you trigger index building after you restore your database.

Restoring Greenplum Chorus

You can restore Greenplum Chorus manually:

1. Reinstall Greenplum Chorus, following the instructions in Chapter 2, “Installing or Upgrading Greenplum Chorus”.
2. Before you start Chorus, restore the configuration and data files from the most recent backup. For example:

```
$ cd <chorus install path>
$ source chorus_path.sh
$ chorus_control.sh restore /data/greenplum-chorus/daily_bu/
            greenplum_chorus_backup_20121108_012809.tar
```

Increasing the Memory of Greenplum Chorus

Chorus runs in the JVM, so the memory available to Chorus is the memory available to the JVM.

1. Edit the <*installation directory*>`/shared/chorus.properties` file. Change the `java_options` entry as you wish. For example:

```
java_options=-Djava.library.path=$CHORUS_HOME/vendor/hadoop/
lib/ -server -Xmx1024m -Xms512m -XX:MaxPermSize=128m
```

2. Restart Greenplum Chorus.

The `-Xmx` variable indicates the maximum memory allocated to the JVM. For example, to reset the maximum memory to 2G, you can change `-Xmx1024M` to `-Xmx2048M`.

The `-Xms` variable indicates the memory allocated to the JVM at startup. For example, to reset the startup memory to 1G, you can change `-Xms512M` to `-Xms1024M`.

Working with Greenplum Chorus Log Files

Log levels

Depending on the log level set in `chorus.properties`, the volume of the log files can vary drastically. Supported log levels are:

- debug
- info
- warn
- error
- fatal

production.log

The rails `production.log` file is stored in:
`<chorus-root>/shared/log/production.log`

This log contains information on requests sent to the Chorus webserver and various debugging information. For example: `server errors`, `file not found`, `permission denied`, and others.

worker.production.log

The rails `worker.production.log` file is stored in:
`<chorus-root>/shared/log/worker.production.log`

It contains logs for the background worker threads that Chorus uses to perform various asynchronous tasks like database imports, checking instance statuses, etc.

scheduler.production.log

The rails `scheduler.production.log` file is stored in:
`<chorus-root>/shared/log/scheduler.production.log`

It contains information about jobs that the scheduler issues to different background workers. This will mainly show that a task was scheduled. See the worker.production.log for more detailed information about what happened during execution of a task.

solr-production.log

The rails `solr-production.log` file is stored in:
`<chorus-root>/shared/log/solr-production.log`

It contains information about solr search queries issued against Chorus.

nginx

`nginx` maintains `access.log` and `error.log` files in
`<chorus-root>/shared/log/nginx`

syslog

As an alternative to the log files listed above, all logs can be combined in one file by using `syslog` as the logger.

To turn on `syslog` as the logger, put `logging.syslog = true` in
`<chorus>/shared/chorus.properties`.

Logrotate

You can use the Linux command `logrotate` to rotate your log files and prevent accumulation. By running `logrotate your_logrotate.conf` from a cron job, you can make sure the logs get rotated at preset intervals.

Here is an example of a `your_logrotate.conf` configuration file that rotates all the important Chorus log files:

```
daily
rotate 4
copytruncate
size 10M
<chorus>/shared/log/production.log {
}
<chorus>/shared/log/nginx/access.log {
}
<chorus>/shared/log/nginx/error.log {
}
<chorus>/shared/log/solr-production.log {
}
<chorus>/shared/log/worker.production.log {
}
<chorus>/shared/log/scheduler.production.log {
```

```
}
```

See the `logrotate` manual page for more details on the features of `logrotate`: http://linuxcommand.org/man_pages/logrotate8.html.

Note: If you use `syslog`, you don't need to rotate your logs manually—`syslog` rotates the log files for you.