

INSTALLING AVALANCHE ON LINUX WITH POSTGRESQL

You can install and manage many of the components of Avalanche MC on a Linux operating system. This includes the Enterprise Server, Statistics Server, Mobile Device Server, License Server, Service Manager, and InfoRail. These components can be installed using RPMs, tar files, or the Mobile Device Server can be deployed from the Avalanche Console after installing the other components.

NOTE: The Avalanche Java Console must be installed on a Windows operating system. Refer to the *Avalanche 5.3 System Requirements* paper on the Wavelink Web site for details on system requirements.

In addition to the Avalanche files provided by Wavelink, you will also need to acquire and install a JRE, Apache Tomcat, and PostgreSQL. More information on these is available in the [Installation Prerequisites on page 2](#).

Perform the following steps in order to ensure the Avalanche servers are correctly installed on a Linux operating system:

- 1 Verify that installation requirements are met and that the JRE and Tomcat are installed. You should also have the database management system installed.
- 2 Set user permissions to allow the user to execute scripts and commands for each server. The user should be set to *not* require TTY for any computer where a mobile device server will be deployed.
- 3 Ensure that traffic between components will not be blocked by firewalls. For information on what ports need to be open to allow Avalanche traffic, see the Ports list in the Avalanche MC User Guide.
- 4 Obtain the installation files and install the Avalanche components.
- 5 Populate and configure the databases and extract the WAR file.
- 6 Start the servers.
- 7 If necessary, increase the open files limit for the InfoRail server and/or the JVM memory settings for Tomcat.
- 8 Install the Avalanche Java Console on a Windows computer.
- 9 Launch the Avalanche Java Console and use it to deploy any Mobile Device Server RPMs.

The following sections provide details on how to install Avalanche on Linux:



- [Installation Prerequisites](#)
- [Installing with RPM Files](#)
- [Populating the Databases](#)
- [Configuring the Database and Tomcat Files](#)
- [Extracting the WAR File for Tomcat](#)
- [Starting the Services](#)
- [Increasing the Open Files Limit](#)
- [Increasing the JVM Memory Parameters for Tomcat](#)

INSTALLATION PREREQUISITES

For system requirements, see the *Avalanche 5.3 System Requirements* paper on the Wavelink Web site.

The following tasks must be done before you install Avalanche on Linux:

- Install Java Runtime Environment (JRE) 1.6.24 or newer. The JRE is required by the Enterprise and Statistics Servers, License Server, InfoRail, Tomcat, and the DBMS.
- Create a user with sufficient rights to install programs and create and maintain the Avalanche Server working directory. The user should be set to *not* require TTY. Also, if you plan to use USB and serial support with Avalanche, the user at the Mobile Device Server must have appropriate permissions.
- Set a sufficiently high limit on open files.
- Install Tomcat 7.0.x at the location you plan to install the Enterprise server.
- Install PostgreSQL 9.1.2.
- Create the Avalanche databases. The databases must be named `avalanche53` and `avastats53`.
- Create the login role for the databases.
- If you are installing the Enterprise Server in a different location than the database server, modify `pg_hba.conf` and `postgresql.conf` to support a remote connection.



INSTALLING WITH RPM FILES

The preferred method of installing Avalanche services uses RPM files. When you install using an RPM, it creates the directories for the files and adds the services to the Service Manager.

NOTE: You also have the option to install using tar files. Create directories for the tar files, and then extract the files to the proper directories.

Wavelink recommends installing the following files at the Enterprise Server location: Enterprise Server, License Server, Statistics Server, InfoRail service, and the Tomcat WAR files.

Wavelink recommends installing the Mobile Device Server and Service Manager together using the Java Console to deploy the server. You also have the option to install the Mobile Device Server and Service manager using RPMs or tar files. If you install using RPMs or tar files, you must configure the server properties files before starting the services. For information on deploying a server, see the *Deploying a Mobile Device Server for Linux* paper on the Wavelink web site.

To install using RPM files:

- 1 Obtain the files from Wavelink.
- 2 From a shell, type:

```
rpm -i [filename].rpm
```

where [filename] is the name of the RPM file you want to install. Repeat this task for all the RPM files you want to install locally.

POPULATING THE DATABASES

After you have created the databases, they need to be populated with Avalanche setup information. The scripts used to populate the databases are included in the Enterprise Server and Statistics Server installation.

If you have the Enterprise Server, Statistics Server, and PostgreSQL databases all on the same machine, running the `db_setup.sh` script will run all the scripts in order.

If PostgreSQL is installed on a different machine than the Enterprise and Statistics Servers, you may need to copy the files from the server installation location to the computer where the databases are. After installing the Enterprise and Statistics Servers, the Enterprise Server scripts are in the `eserver/db/Postgres` directory:

```
avalanche.sql  
quartz.sql  
amc_en_US_CREATE_HB_SEQ.sql  
create_sequences.sql  
amc_en_US_INSERT_DEFAULT.sql
```

And the Statistics Server scripts are in the `statserver/db/Postgres` directory:

```
avastats.sql
stat_en_US_CREATE_HB_SEQ.sql
stat_en_US_INSERT_DEFAULT.sql
```

You may need to copy the files from the server installation location to the computer where the databases are before running the scripts. The scripts can be run from a command line.

To populate local PostgreSQL databases:

1 Navigate to `/opt/wavelink/eserver/db`

2 Open the `db_setup.sh` script and find the following section:

```
export DBNAME=avalanche53;
export PGUSER=postgres;
export PGPASS=admin23
export PGPASSWORD=$PGPASS
export DBHOST=127.0.0.1;
export DBPORT=5432;
```

3 Change the `DBNAME`, `PGUSER`, `PGPASS`, `DBHOST`, and `DBPORT` values to the database name, username, password, host address, and port for your Enterprise Server database. Save your changes.

4 From the `eserver` directory, type `sudo db/db_setup.sh` to run the script that populates the Enterprise Server database.

5 Navigate to `/opt/wavelink/statserver/db`

6 Open the `db_setup.sh` script and find the following section:

```
export DBNAME=avalanche53;
export PGUSER=postgres;
export PGPASS=admin23
export PGPASSWORD=$PGPASS
export DBHOST=127.0.0.1;
export DBPORT=5432;
```

7 Change the `DBNAME`, `PGUSER`, `PGPASS`, `DBHOST`, and `DBPORT` values to the database name, username, password, host address, and port for your Statistics Server database. Save your changes.

8 From the `statserver` directory, type `sudo db/db_setup.sh` to run the script that populates the Statistics Server database.

To populate remote PostgreSQL databases:

1 Copy the scripts files to the PostgreSQL `bin` directory.

2 Use the following commands from a command line to run the scripts in order, replacing `[path]` with the path to the script:

```
psql.exe -U postgres -d avalanche53 -f [path]\avalanche.sql
psql.exe -U postgres -d avalanche53 -f [path]\quartz.sql
```

```
psql.exe -U postgres -d avalanche53 -f [path]\amc_en_US_CREATE_HB_SEQ.sql
```

```
psql.exe -U postgres -d avalanche53 -f [path]\create_sequences.sql
```

```
psql.exe -U postgres -d avalanche53 -f [path]\amc_en_US_INSERT_DEFAULT.sql
```

```
psql.exe -U postgres -d avastats53 -f [path]\avastats.sql
```

```
psql.exe -U postgres -d avastats53 -f [path]\stat_en_US_CREATE_HB_SEQ.sql
```

```
psql.exe -U postgres -d avastats53 -f [path]\stat_en_US_INSERT_DEFAULT.sql
```

3 Navigate to `/opt/wavelink/eserver/db`

4 Open the `db_setup.sh` script and find the following section:

```
export DBNAME=avalanche53;
export PGUSER=postgres;
export PGPASS=admin23
export PGPASSWORD=$PGPASS
export DBHOST=127.0.0.1;
export DBPORT=5432;
```

5 Change the `DBNAME`, `PGUSER`, `PGPASS`, `DBHOST`, and `DBPORT` values to the database name, username, password, host address, and port for your Enterprise Server database.

6 Delete the following section:

```
# Make an honest attempt to add the postgres installation directory to
# the path. Unfortunately, this is not going to work for all distros.

if [ -e /opt/PostgreSQL/9.1/bin/psql ]; then
PATH=$PATH:/opt/PostgreSQL/9.1/bin
fi
if [ -e /opt/PostgreSQL/9.0/bin/psql ]; then
PATH=$PATH:/opt/PostgreSQL/9.0/bin
fi
if [ -e /opt/PostgreSQL/8.4/bin/psql ]; then
PATH=$PATH:/opt/PostgreSQL/8.4/bin
fi

echo "Initializing $SCRIPTPATH database"

dropdb -h $DBHOST $DBNAME
createdb -h $DBHOST $DBNAME
psql -q -h $DBHOST -d $DBNAME -f ./db/$SCRIPTPATH/amc_en_US_CREATE_HB_SEQ.sql
psql -q -h $DBHOST -d $DBNAME -f ./db/$SCRIPTPATH/create_sequences.sql
psql -q -h $DBHOST -d $DBNAME -f ./db/$SCRIPTPATH/avalanche.sql
psql -q -h $DBHOST -d $DBNAME -f ./db/$SCRIPTPATH/quartz.sql
psql -q -h $DBHOST -d $DBNAME -f ./db/$SCRIPTPATH/amc_en_US_INSERT_DEFAULT.sql
```

7 Save your changes.

8 From the `eserver` directory, type `sudo db/db_setup.sh`

CONFIGURING THE DATABASE AND TOMCAT FILES

In order for the Enterprise Server, Statistics Server, and Web Console to communicate with the databases, some files need to be configured with the database information while the servers are not running.

These files are in the `eserver/conf/main` directory. Once you have modified the files for the Enterprise Server, copy them to the directories for the Statistics Server and Tomcat.

To modify these files, you must know the address of the database computer, the port the databases are using, the database names, username, and password.

To configure the dao and quartz properties files:

1 On the computer where you installed the Enterprise Server, navigate to the `eserver/conf/main` directory and open the `dao-factory.xml` file with a text editor.

2 Near the beginning of the file, locate the section with the following properties:

```
<property name="url" value="jdbc:postgresql://localhost:5432/avalanche53"/>
<property name="username" value="postgres"/>
<property name="password" value="Admin!23"/>
```

3 Change the computer address (`localhost`), port (`5432`), Enterprise Server database name (`avalanche53`), username (`postgres`), and password (`Admin!23`) values to the values for the Enterprise Server.

4 Save and close the file.

5 Open `dao-factory-ss-es.xml` and repeat steps 2 through 4.

6 Open `dao-factory-ss.xml` and repeat steps 2 through 4, replacing the default values with the values for the Statistics Server.

7 Open `quartz.avaserver.properties` and locate the lines at the end with the following properties:

```
org.quartz.dataSource.mme.URL =
jdbc:postgresql://localhost:5432/avalanche53
org.quartz.dataSource.mme.user = postgres
org.quartz.dataSource.mme.password = Admin!23
```

8 Change the computer address (`localhost`), port (`5432`), Enterprise Server database name (`avalanche53`), username (`postgres`), and password (`Admin!23`) values to the values for the Enterprise Server.

- 9 Save and close the file.
- 10 Copy the `dao-factory.xml`, `dao-factory-ss.xml`, and `dao-factory-ss-es.xml` files to the computer where the Statistics Server is installed and put them in the `statserver/conf/main` directory. This will overwrite the existing files with the same names.
- 11 Copy the `dao-factory.xml`, `dao-factory-ss.xml`, and `dao-factory-ss-es.xml` files to the computer where Tomcat is installed and put them in the Tomcat `lib` directory.

EXTRACTING THE WAR FILE FOR TOMCAT

In order to use the Web Console with Avalanche, you must have Tomcat installed on the same box as the Enterprise Server. You must have the Enterprise Server installed in order to have the WAR file.

To extract the WAR file for Tomcat:

- 1 Ensure that the Tomcat service is not running.
- 2 Copy the WAR file (included in the Enterprise Server installation zip file) to the `tomcat/webapps` directory.
- 3 Restart Tomcat. Tomcat extracts the WAR file.

STARTING THE SERVICES

Once the files are installed and configured, start the services. The following method uses a command line, or you can use another method if you prefer. You will need to start the Enterprise Server, Statistics Server, License Server, and InfoRail. Tomcat should also be started up after the other services.

NOTE: If you deploy the Mobile Device Server from the Avalanche Console, you do not need to manually start the service.

To start the services from a command line:

- 1 Use the following commands:

```
Sudo /etc/init.d/wl-inforail start
Sudo /etc/init.d/wl-eserver start
Sudo /etc/init.d/wl-licserver start
Sudo /etc/init.d/wl-statserver start
```
- 2 Navigate to the `tomcat/bin` directory and use the following command to start Tomcat:

```
Sudo ./startup.sh
```

INCREASING THE OPEN FILES LIMIT

If the InfoRail server is run under a user account whose per-process limit on open files is too low, the log file returns entries with errors. New subscribers will not be able to connect to the router. To raise the number of open connections or files, the per-process limit needs to be increased. The limit can be increased temporarily using the `ulimit` command before executing the router or permanently by increasing the limit for the account or the whole system.

You may also receive error messages if the overall number of open files for the entire system has been exceeded. You can check the configured limit for open files and then increase the overall number of open files for the system.

To temporarily increase:

- 1 Insert this command into the script that starts the router:

```
ulimit -n <new limit>
```

- 2 Log out of the system and then log back in for the changes to take effect.

To permanently increase:

- 1 Add the following lines to the `etc/security/limits.conf` file:

```
<account> soft nofile <softlimit>
```

```
<account> hard nofile <hardlimit>
```

The account field can be in various forms such as group names and wild cards.

- 2 Log out of the system and then log back in for the changes to take effect.
- 3 Modify the startup scripts for each of the device servers that need increased open file limits.

To check the configured limit:

- From a shell, enter the following command:

```
cat /proc/sys/fs/file-max
```

This command reports three values. The first value is the total allocated file openings. The second value is the total free allocated file openings. The third value is the maximum open files allowed.

To increase the file-max value:

- 1 Navigate to `/etc/sysctl.conf` and add the following line:

```
fs.file-max = <new limit>
```

- 2 Reboot the system.

INCREASING THE JVM MEMORY PARAMETERS FOR TOMCAT

If Tomcat is overloaded, it may throw an Out of Memory error (`OutOfMemoryError: Java heap space`). Examples of events that could cause this are using an extensive number of instances of the Web Console simultaneously or running lots of reports. Increasing the memory parameters for Tomcat can prevent this error. When you increase the parameters, take into consideration the demands on Tomcat and the resources available.

To increase the JVM memory parameters:

- 1 Navigate to:

```
[installation directory]/AvalancheMC/WebUtilities/Tomcat/bin
```

- 2 Open the file `catalina.sh` in a text editor.

- 3 Insert the following line before `# OS specific support`:

```
CATALINA_OPTS="-Xmx512m -XX:MaxPermSize=256m -XX:PermSize=256m"
```

The first argument will set the available memory for data while the second and third arguments will set the PermGen size (available memory for code; e.g., Java class files). Change the memory parameters depending on system demands and resources available.

- 4 Restart the Apache Tomcat for Wavelink service.



Wavelink Corporation

USA and Canada: 1.888.697.WAVE (9283)

Outside the USA and Canada: + 800 WAVELINK (9283 5465)

www.wavelink.com

