JAMF Software Server Installation Guide for Linux

Version 8.6

Contents

Overview

What's in This Guide

This guide explains how to install and upgrade the JSS using the JSS Installer for Linux. It also explains how to back up and restore the database, and troubleshoot the JSS.

Additional Resources

For information on related topics, see the following resources:

- Product Administrator's Guides—For information about setting up and managing the JSS after it is installed, refer to the Administrator's Guide for your product. The Casper Suite, Imaging Suite, and Recon Suite Administrator's Guides are available at:

  http://jamfsoftware.com/resources/documentation

- "Manually Installing the JAMF Software Server"— If you do not want to use the JSS Installer, you can follow the instructions in this technical paper to manually install the JSS on Linux or Windows platforms. You can download it from:

  http://jamfsoftware.com/libraries/pdf/white_papers/Manually_Installing_the_JAMF_Software_Server.pdf

Requirements

This section lists the requirements for the applications and utilities you need to install and maintain the JAMF Software Server (JSS).

JAMF Software Server

You can host the JSS on any server that meets the following minimum requirements:

- Java 1.6
- MySQL 5.1 or later
- Apache Tomcat 6.0 or later

Tested Linux operating systems include:

- Ubuntu 10.04 LTS Server
- Ubuntu 12.04 LTS Server
- Red Hat Enterprise Linux (RHEL) 6

Note: To install the JSS using the JSS Installer for Linux, the server must meet the installer requirements.

JSS Installer for Linux

The JSS Installer for Linux requires a server with:

- A 64-bit capable Intel processor
- 2 GB of RAM
- 400 MB of disk space available
- One of the following operating systems:
    - Ubuntu 10.04 LTS Server (64-bit)
    - Ubuntu 12.04 LTS Server (64-bit)
    - Red Hat Enterprise Linux (RHEL) 6
- Open Java Development Kit (OpenJDK) 6

    For more information, go to http://openjdk.java.net/.
- MySQL Enterprise Edition 5.5 or later (recommended) *or* MySQL Community Server 5.5 or later, available at:

    https://www.mysql.com/downloads/
- Ports 8443 and 8080 available

JSS Database Utility

The JSS Database Utility requires a server with MySQL Server 5.1 or later.

Installing the JSS

Installing the JAMF Software Server (JSS) involves the following steps:

1. Install the required software (if you haven't already).

2. Create the jamfsoftware database.

3. Run the JSS Installer.

   This section includes details for each step.

   Before you begin, review the "Requirements" section and make sure that your server meets the JSS Installer requirements.

Step 1: Install the Required Software

Java and MySQL must be installed on the server before you can create the jamfsoftware database and run the JSS Installer. For instructions on how to install and configure Java and MySQL, see the following Knowledge Base article:

https://jamfnation.jamfsoftware.com/article.html?id=28

Step 2: Create the jamfsoftware Database

Create a MySQL database in which the JSS can store its data, and a MySQL user that can access it. Name the database "jamfsoftware" and give the MySQL user the following credentials:

- Username: jamfsoftware
- Password: jamfsw03

> **Note:** If you choose to customize the database name, username, or password, you will need to enter the custom settings after running the JSS Installer.

To create the jamfsoftware database:

1. Open a command prompt and access the MySQL command-line utility as "root" by typing something similar to the following:

```
mysql -u root -p
```

   If you did not create a root password when installing MySQL, do not type `-p`.
   If MySQL is not in the path or it is installed in a custom location, enter the installation path in place of `mysql`.

2. If prompted, enter the password for the MySQL "root" user.

3. Create a database named "jamfsoftware" by executing:

```
CREATE DATABASE jamfsoftware;
```

4. Grant permissions to a MySQL user named "jamfsoftware" so that it can access the new database:

```
GRANT ALL ON jamfsoftware.* TO 'jamfsoftware'@localhost IDENTIFIED BY
    'jamfsw03';
```

> *Note:* If you choose to enter a username other than "jamfsoftware," it is recommended that you do not use "root".

Step 3: Run the JSS Installer

Run the JSS Installer to install Apache Tomcat and the JSS web application.

To run the JSS Installer:

1. Copy the JSS Installer for Linux (`jssinstaller.run`) to the server.

2. Log in to the server as a user with superuser privileges.

3. Initiate the installer by executing:

```
sudo sh /path/to/jssinstaller.run
```

4. When the requirement check is complete, type "y" to proceed.

5. (RHEL only) When the installation is complete, edit the firewall configuration to allow access to port 8443 by executing:

```
sudo system-config-firewall-tui
```

6. (RHEL only) Choose **Other** or **Customize**, and then manually add port 8443 with TCP protocol. The option you choose depends on whether you have a GUI or shell-only interface.

7. Access the JSS from a client machine by opening a web browser and typing the protocol, IP address or DNS name of the server, and port. For example:

    https://jss.mycompany.com:8443/

8. If you entered a custom database name, username, or password when you created the jamfsoftware database, or MySQL is using a port other than 3306, the Database Connection Properties pane is displayed. Before you can access the JSS, you must follow the onscreen instructions to create a connection between the JSS and the jamfsoftware database.

Installed Files and Folders

The following files and folders are installed on the server when you run the JSS Installer:

Apache Tomcat
Tomcat is the web application server that runs the JSS web application. A directory named `tomcat` is installed at:

`/usr/local/jss/tomcat/`

jamf.tomcat7
This is the service file for Tomcat. It is installed in the following location:

`/etc/init.d/jamf.tomcat7`

Database backup location
The JSS Database Utility stores database backups in the following location by default:

`/usr/local/jss/backups/database/`

JSS Database Utility
The JSS Database Utility is installed in the following location:

`/usr/local/jss/bin/JSSDatabaseUtil.jar`

JSS web application
The JSS is a web application that runs on Tomcat. A directory named ROOT is installed at:

`/usr/local/jss/tomcat/webapps/ROOT/`

keystore
Tomcat requires a .keystore file to provide connections over SSL. The JSS Installer creates a default .keystore file and stores it in the following location:

`/usr/local/jss/tomcat/.keystore`

Logs
Logs for the installation and for the JSS are stored in the following directory:

`/usr/local/jss/logs/`

server.xml
The JSS Installer installs a modified copy of Tomcat's `server.xml` file. This file enables SSL, ensures that the JSS appears in the `root` context, and enables database connection pooling. It is installed in the following location:

`/usr/local/jss/tomcat/conf/server.xml`

Allocating Additional Memory to Tomcat

This section explains how to:

- View the amount of memory being used by the web application
- Allocate additional memory to Tomcat

To view web application memory usage:

1. Log in to the JSS with a web browser.

2. Click the **Settings** tab.

3. Click the **Database/Web Application Health** link.

4. Click the **Web App Memory** link.

To allocate additional memory to Tomcat using the JSS Database Utility:

1. Open the JSS Database Utility on the server running the JSS by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
```

2. If prompted, enter the password for your account and then press the Return key.

3. If the JSS Database Utility is unable to locate the MySQL binary, you are prompted to enter the path. Click **Continue** and enter the location of the binary.

4. From the menu bar, choose **Utilities** > **Change Tomcat settings**.

5. Modify the minimum and maximum memory and PermGen sizes as needed.
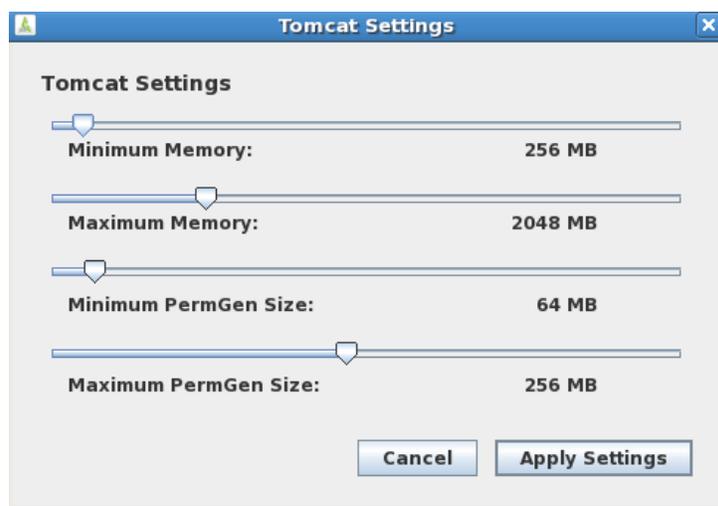
6. Click **Apply Settings**.

7. When prompted to restart Tomcat, click **Yes**.

JSS Database Utility

? Would you like the JSS Database Utility to restart Tomcat?

Yes   No

Upgrading the JSS

This section explains how to upgrade the JAMF Software Server (JSS).

> **Note:** The JSS Installer cannot be used to upgrade the JSS v8.1 or earlier.

To upgrade the JSS:

1. Back up the current database using the JSS Database Utility. (See "Backing Up the Database" for complete instructions.)

2. Back up custom reports.

   If you used the JSS Installer to install the JSS, custom reports are located in `/usr/local/jss/tomcat/webapps/ROOT/WEB-INF/reporting/`.

3. Copy the most current version of the JSS Installer for Linux (`jssinstaller.run`) to the server.

   To obtain the most current installer for Linux, see the introductory email that you received from JAMF Software or contact your JAMF Software Representative.

4. Log in to the server as a user with superuser privileges.

5. Initiate the installer by executing:

```
sudo sh /path/to/jssinstaller.run
```

6. Follow the onscreen instructions to complete the upgrade.

Backing Up the Database

The JSS Database Utility allows you to create backups of the jamfsoftware database, schedule database backups, and stop scheduled database backups.
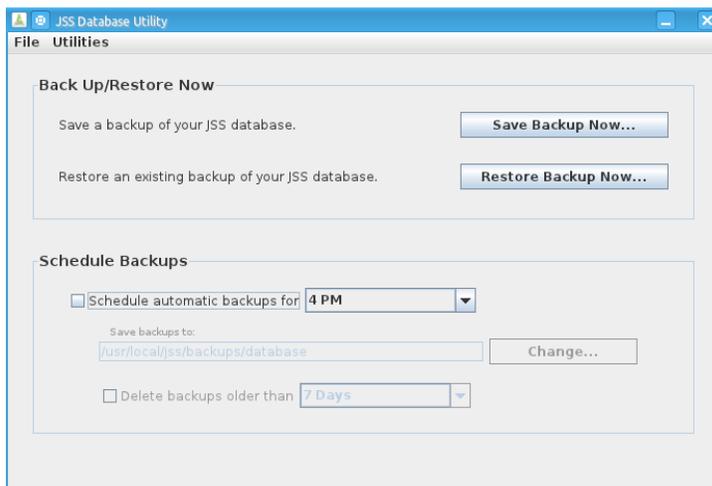
Creating Database Backups

Use the JSS Database Utility to create a backup of the jamfsoftware database. The time it takes to create the backup depends on the size of the database.

To create a database backup using the GUI:

1. Open the JSS Database Utility by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
```

2. If prompted, enter the password for your account and then press the Return key.

3. If the JSS Database Utility is unable to locate the MySQL binary, you are prompted to enter the path. Click **Continue** and enter the location of the binary.

4. If the Database Connection Setup pane appears, edit the settings to match your database configuration and click **Apply Settings**.

5. Click **Save Backup Now**.

6. Select the location where you want to save the backup, and then click **Choose**.

The JSS Database Utility creates the backup and saves it as a .sql.gz file.

To create a database backup from the command line:

1. Execute the following command:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar backup
    -saveBackupTo <path>
```

For example:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar backup
    -saveBackupTo /home/username/backups/
```

2. If prompted, enter the password for your account and then press the Return key.

3. If prompted, edit the database configuration settings to match your own by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar backup
    -saveBackupTo <path> -server <database server address> -port <database
    port> -database <database name> -user <database user> -pass
```

For example:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar backup
    -saveBackupTo /home/username/backups/ -server localhost -port 3306
    -database jamfsoftware -user jss -pass
```

The JSS Database Utility creates the backup and saves it as a .sql.gz file.

To access command-line help for the $backup$ verb, execute:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar help backup
```

Scheduling Database Backups

Use the JSS Database Utility to schedule daily backups of the jamfsoftware database. You can also automate the deletion of scheduled backups that are older than a certain number of days.

To schedule database backups from the GUI:

1. Open the JSS Database Utility by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
```

2. If prompted, enter the password for your account and then press the Return key.

3. If the JSS Database Utility is unable to locate the MySQL binary, you are prompted to enter the path. Click **Continue** and enter the location of the binary.

4. If the Database Connection Setup pane appears, edit the settings to match your database configuration and click **Apply Settings**.

5. Select the **Schedule automatic backups for** checkbox and choose the hour of the day that you want backups to occur.

6. To change the location where backups are saved, click the **Change** button and select a new location.
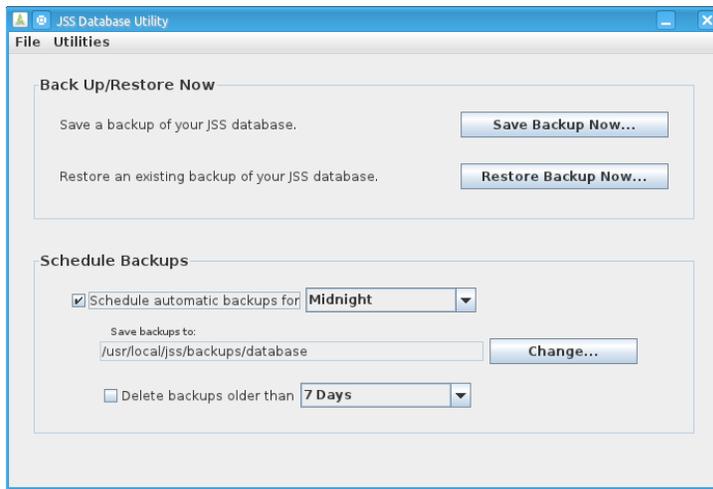
7. To automate the deletion of scheduled backups, select the **Delete backups older than** checkbox. Then, choose the number of days after which backups should be deleted.

   The JSS Database Utility saves daily backups at the hour that you specified. It also deletes scheduled backups older than the number of days that you specified.

   To schedule database backups from the command line:

1. Execute the following command:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar schedule
   -hour <number between 0 and 23> -scheduledBackupPath <path>
   -deleteBackupsOlderThanDays <number greater than 0>
```
   For example:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
   schedule -hour 3 -scheduledBackupPath /home/username/backups/
   -deleteBackupsOlderThanDays 7
```

2. If prompted, enter the password for your account and then press the Return key.

3. If prompted, edit the database configuration settings to match your own by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar schedule
   -hour <number between 0 and 23> -scheduledBackupPath <path>
   -deleteBackupsOlderThanDays <number greater than 0> -server <database
   server address> -port <database port> -database <database name> -user
   <database user> -pass
```

For example:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
   schedule -hour 3 -scheduledBackupPath /home/username/backups/
   -deleteBackupsOlderThanDays 7 -server localhost -port 3306 -database
   jamfsoftware -user jss -pass
```

The JSS Database Utility saves daily backups at the hour that you specified. It also deletes scheduled backups older than the number of days that you specified.

To access command-line help for the `schedule` verb, execute:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar help schedule
```
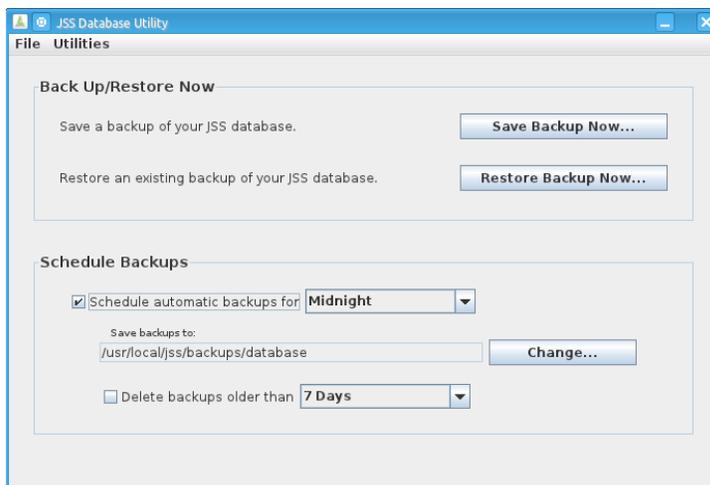
Stopping Scheduled Database Backups

Use the JSS Database Utility to stop scheduled backups of the jamfsoftware database.

To stop scheduled database backups using the GUI:

1. Open the JSS Database Utility by executing:

    ```
    sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
    ```

2. If prompted, enter the password for your account and then press the Return key.

3. If the JSS Database Utility is unable to locate the MySQL binary, you are prompted to enter the path. Click **Continue** and enter the location of the binary.

4. If the Database Connection Setup pane appears, edit the settings to match your database configuration and click **Apply Settings**.

5. Deselect the **Schedule automatic backups for** checkbox.

    The JSS Database Utility stops scheduled backups immediately.

To stop scheduled database backups from the command line:

Execute the following command:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar unschedule
```

The JSS Database Utility stops scheduled backups immediately.

To access command-line help for the `unschedule` verb, execute:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar help unschedule
```

Restoring Database Backups

If you need to revert to an earlier version of your database, you can use the JSS Database Utility to restore a database backup.

To restore a database backup using the GUI:

1. Open the JSS Database Utility by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar
```

2. If prompted, enter the password for your account and then press the Return key.

3. If the JSS Database Utility is unable to locate the MySQL binary, you are prompted to enter the path. Click **Continue** and enter the location of the binary.

4. If the Database Connection Setup pane appears, edit the settings to match your database configuration and click **Apply Settings**.

5. Click **Restore Backup Now**.
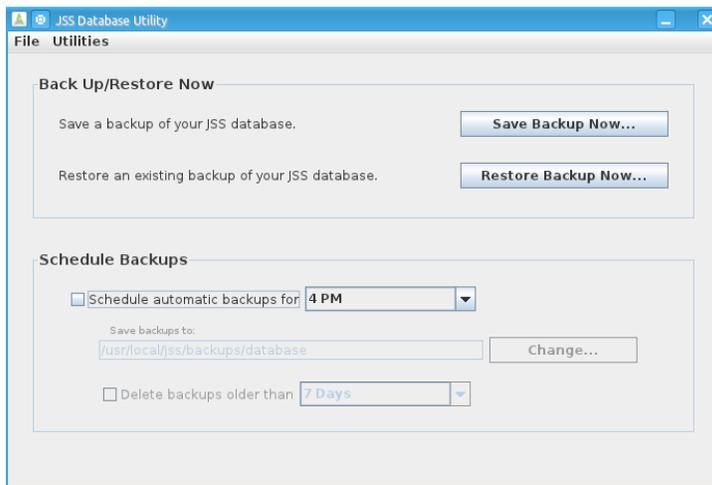
6. Select the backup that you want to restore (.sql or .sql.gz), and then click **Choose**.

7. When prompted to restart Tomcat, click **Yes**.

The JSS Database Utility restarts Tomcat and replaces the current database with the one that you restored.

To restore a database backup from the command line:

1. Execute the following command:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar restore -file
    <filename>
```

For example:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar restore -file backup.
    sql.gz
```

2. If prompted, enter the password for your account and then press the Return key.

3. If prompted, edit the database configuration settings to match your own by executing:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar restore -file
    <filename> -server <database server address> -port <database port>
    -database <database name> -user <database user> -pass
```

For example:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar restore -file backup.
    sql.gz -server localhost -port 3306 -database jamfsoftware -user jss
    -pass
```

4. Restart Tomcat by executing:

```
sudo /etc/init.d/jamf.tomcat7 restart
```

The JSS Database Utility replaces the current database with the one that you restored.

To access command-line help for the `restore` verb, execute:

```
sudo java -jar /usr/local/jss/bin/JSSDatabaseUtil.jar help restore
```

Troubleshooting the JSS

Most issues with the JSS have to do with the configuration of Apache Tomcat or MySQL. The JSS Database Utility allows you to troubleshoot most of the issues that you encounter.

This section explains how to troubleshoot the following issues:

- Connection issues
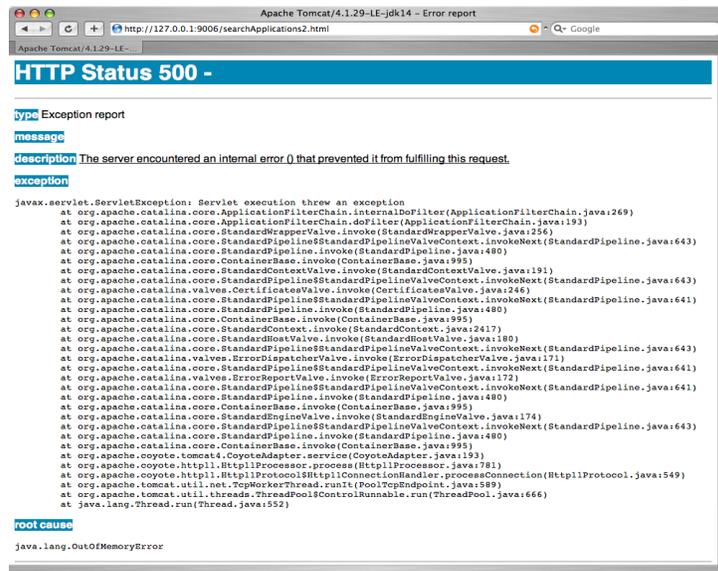- Memory issues
- Database issues

Connection Issues

If applications are not connecting to the JAMF Software Server (JSS), you can use a web browser to troubleshoot the issue. If you are able to connect to the JSS, the applications should be able to connect as well.

To troubleshoot connection issues:

1. Open a web browser and try connecting to the JSS on port 8443.

    For example, if the DNS name of the JSS is "jss.mycompany.com", try connecting to:

    https://jss.mycompany.com:8443/

2. If you are prompted to verify a certificate, accept the certificate.

3. If you are able to connect to the JSS, make sure that the application is pointing at the correct IP address.

    a. Quit the application.

    b. Hold down the Alt key and re-open the application to bring up the Preferences pane.

    c. On the Preferences pane, enter the DNS name or IP address for the JSS and then click **Save**.

    d. Enter the username and password for an administrator account to the server, and then click **OK**.

4. If the application still fails to connect, restart Tomcat.

    For instructions on how to restart Tomcat, see the Knowledge Base article at:

    https://jamfnation.jamfsoftware.com/article.html?id=117

Memory Issues

If there is a large amount of data in the JSS, you may need to allocate additional memory to Tomcat. Tomcat displays the following error page if more memory is required:

For instructions on viewing the amount of memory being used by the web application and allocating additional memory to Tomcat, see "Allocating Additional Memory to Tomcat".

Database Issues

MySQL database tables can become corrupt if the JSS is running on a very slow computer to which many clients are connected, or if the server running the JSS crashed and the database was not shut down properly.

When errors occur in the MySQL database, the JSS displays an alert similar to the following:

Got error 127 from table handler

This section explains how to:
- View the status of database tables
- Repair database tables
- Optimize database tables

Viewing the Status of Database Tables

To view the status of database tables:

1. Log in to the JSS with a web browser.

2.	Click the **Settings** tab.

3.	Click the **Database/Web Application Health** link.

4.	Click the **Database Table Status** link.

Repairing Database Tables

If you have a large database, it may take longer to verify the status of your database tables. Once the status of each table is returned, you may want to repair the tables that do not return an "OK" status.

To repair database tables:

1.	Log in to the JSS with a web browser.

2.	Click the **Settings** tab.

3.	Click the **Database/Web Application Health** link.

4.	Click the **Repair Database Tables** link.

Optimizing Database Tables

Optimizing database tables allows you to ensure that each table's index is up to date so that you can perform database lookups as quickly as possible.

To optimize database tables:

1.	Log in to the JSS with a web browser.

2.	Click the **Settings** tab.

3.	Click the **Database/Web Application Health** link.

4.	Click the **Optimize Database Tables** link.