

The background of the slide is a dark blue gradient with a diagonal line separating a lighter blue area on the left from a darker blue area on the right. The Jive logo is positioned in the lower-left quadrant.

jive

work better together™

Upgrading

Contents

Upgrading.....	3
Before You Upgrade.....	3
Back Up Your jiveHome Directory.....	3
Back Up Your Database.....	3
Remove Plugins Before Upgrading.....	3
For 5.0 and 6.0 Upgrades, Prepare for the New Jive CLI Installer and Interface.....	4
For Pre-6.0 Upgrades, Add a Node for On-Premise Search (Optional).....	5
For Pre-5.0 Upgrades, Add an Activity Engine Node and Database.....	5
Allow Time for Search Index Rebuilding.....	5
Upgrades That Cause Search Index Rebuilds.....	5
Database Upgrades.....	5
Creating a Test Environment for Upgrading.....	6
Upgrading a Linux Package.....	6
Upgrading the Package.....	7
Upgrading Custom Themes and Templates.....	9

Upgrading

This section includes information about upgrading the application.

Important Notes

- Some Jive modules you've purchased have their own system and installation requirements. See the separate documentation for each module. The latest versions of Jive-supported plugins are available from [the Jive Community](#).
- Jive Software supports upgrades from Jive version 4.5 and later.
- The instructions in this section assume you've already installed the application on the target host. If you haven't, use the [installation instructions](#) to do so.

Before You Upgrade

Before you upgrade your instance, take the steps described here (such as backing up important data) to help ensure that your upgrade is successful.



Caution: You must be upgrading from version 4.5.0 or later.

Back Up Your jiveHome Directory

The jiveHome directory contains important information about your environment. The database connection information is stored there, as well as logs, cached attachments, your license file, and the local system database files (if used). You should back up this directory before upgrading.

Back Up Your Database

You should back up your database before you upgrade. For now, the best way to manage database backups is to follow the recommendations of your DBA or the recommendations of your database software. There are a number of tools built in to various databases. Here are a couple of examples:

- MySQL — Use the `mysqldump` tool
- Postgres — Use the `pg_dump` tool

Remove Plugins Before Upgrading

Before starting your upgrade, be sure to remove any plugins you've installed. For those plugins that aren't compatible with the version you're upgrading to, you'll need to separately upgrade your plugin code (or get upgraded versions of the plugins from their developer), then install the upgraded versions after you've completed your Jive upgrade. The latest versions of Jive-supported plugins are available from [the Jive Community](#).

For more on managing plugins, see [Adding and Removing Plugins](#).

For 5.0 and 6.0 Upgrades, Prepare for the New Jive CLI Installer and Interface

The following list will help you prepare for upgrading from pre-7.0 releases.

Jive now enforces a strict policy on HTTPS certificate checking

Jive now enforces a strict policy on HTTPS certificate checking, and will not accept self-signed certificates without you adding them to the application Java keystore. If you are upgrading from Jive 6.0.3 and earlier, and you encounter communication errors between the self-signed application and the Jive server, you need to add the certificate to the Java keystore.

For more on SSL Support, see [Installing and Configuring SSL Support](#). Here's an example of how to install a certificate:

1. Download the certificate from the server and save the resulting certificate to a file called `sam.cert`:

```
openssl s_client -connect yourserver.example.com:443 -showcerts > sam.cert
```

2. Import the certificate into the local keystore on the Jive server:

```
/usr/local/jive/java/jre/bin/keytool -import -alias jiveSamCert -file  
sam.cert -keystore /usr/local/jive/java/jre/lib/security/cacerts
```

Use "jive setup" to move your existing configurations into the new unified format

Move your existing startup configurations into the new unified format. Using the "jive setup" command will provide specific suggestions for how to do this. Allow some time during your upgrade for going through these suggestions. As with all upgrades, it is best practice to test on a staging instance first.

Note that HTTPD now runs on 8080, which is a non-privileged port

Remember that the jive-managed Apache HTTPD service now must run on a non-privileged port, which is port 8080 by default. Use the following instructions to make modifications with or without a load balancer:

Production site with a load balancer:

Change your load balancer's configuration to point to each webapp node's port 8080 instead of port 80.



Note: Note that SSL is now required on a load balancer.

Upgrading a single-node test instance without a load balancer:

Modify the jiveURL in the Admin Console before you upgrade by going to Admin Console: **System > Management > System Properties** and modifying the jiveURL system property. For example, if your test instance's jiveURL was previously set to `http://communitytest.mycompany.com`, set it instead to `http://communitytest.mycompany.com:8080`. Then go ahead with your upgrade.

For Pre-6.0 Upgrades, Add a Node for On-Premise Search (Optional)

If you're upgrading from a version prior to 6.0 and you have chosen to implement on-premise search instead of [Cloud Search](#), you will need to add a node for the Search service. Make sure it meets the requirements specified in [Hardware Requirements](#).

For Pre-5.0 Upgrades, Add an Activity Engine Node and Database

If you're upgrading from a version prior to 5.0, you will need to add a node and database for the Activity Engine. Be sure to read the [Setting Up the Activity Engine](#) topics and [Setting Up a New Database](#) before you begin upgrading.

Allow Time for Search Index Rebuilding

Jive Software recommends that you first test the upgrade on a UAT instance to determine how long the search index rebuild will take. In some cases, it may take days to rebuild the index, depending on the size and activity of the community. Communities with large amounts of content and activity should allow adequate time for the search index rebuild before deploying an upgrade to production.

Upgrades That Cause Search Index Rebuilds

Knowing which upgrades cause the Content and/or User Search indexes to rebuild may help you better plan your upgrade.

The following list tells you which versions of Jive require a Search index rebuild, which can take a significant amount of time for large data sets.

Upgrade Path	Does Jive Force a Search Index Rebuild?
5.x to 6.x	Yes. Jive forces both the Content and User Indexes to rebuild.
6.0.0 to 6.0.1	No
6.0.1 to 6.0.2+	Yes. Jive forces only the Content Index to rebuild.
6.x to 7.x	Yes. Jive forces both the Content and User Indexes to rebuild.
Any version to 8.x	Yes. Jive forces both the Content and User Indexes to rebuild.

Database Upgrades

You'll need to upgrade your database when you upgrade the rest of the installation.

For information about database upgrades, see [Upgrading a Database](#). You may want to review the rest of the best practices in [Database Configuration and Best Practices](#), especially if you are moving to a new database platform.

Creating a Test Environment for Upgrading

Before Upgrading

If you are upgrading an existing instance of Jive, we strongly recommend you create a test environment for the new version before migrating your current database. The following instructions assume that your application name is "sbs" and you are using a Postgres database.

1. Dump the production database and copy it to your target server.

```
pg_dump sbs -Fc -O -U postgres > sbs_prod.dmp
```

2. Re-create the target database.

```
createdb sbs -E UTF-8 -O sbs -U postgres
```

3. Restore the database from production into the target.

```
pg_restore -d sbs -O -x -i -U sbs < /home/your_name/sbs_prod.dmp
```

4. Log into the database and override all of the email addresses to prevent spamming. We strongly recommend disabling email so that your Jive test instance does not spam users with email.

```
update jiveuser set email = username || 'discard@localhost';
```

5. Start the target app server.
6. Rebuild the user and main search indexes (**Admin Console: System > Settings > Search > Index Tasks**).
7. Update the jiveURL system property (**Admin Console: System > Management > System Properties**).
8. Change/disable the Email Monitor (**Admin Console: System > Settings > Email Server > Incoming Email**).
9. Change the Analytics database settings, if applicable.
10. Change your SSO settings, if applicable.
11. Change your video keys, if applicable.

Upgrading a Linux Package

With a new package distribution in hand, you can upgrade your existing package.



Note: Before you upgrade, uninstall all your plugins. You'll need to re-install them via the Admin Console after you have successfully upgraded.

What You'll Need

- Your existing installation of the Jive package.
- The new Jive package you're upgrading to.
- SSH access to the host computer so you can copy the package there for installation.

- Ability to become the root user on the host where the installation is performed, commonly via SSH.

Upgrading the Package

For more detail about the commands, refer to [Installing the Linux Package](#).



Note: Executing the `rpm` command to upgrade the package will automatically stop Jive applications. The application will be started again at the end of the upgrade.

1. Be sure to read the [system requirements](#) for important information about software, hardware, and network requirements and recommendations.
2. Make sure you have read [Before You Upgrade](#) and performed the relevant steps listed there.
3. [Remove all plugins](#) from your existing instance. (You'll need to re-install them via the Admin Console after you've successfully upgraded). For a list of supported Jive plugins, see [this list in the Jive Community](#).
4. Remove custom WARs and fixpacks that might exist as provided by support.
5. If you are upgrading from a version later than 5.0, if possible, pull the web application server(s) from the load balancing pool to prevent new activity coming in. Make sure your Activity Engine queues are drained to zero. You can see the queues in **Admin Console: System > Settings > Activity Engine**. (Versions before 5.0 do not use an Activity Engine.)
6. As a jive user, stop Jive services on all of the nodes, beginning with the web application nodes, and stopping the cache services last.



Note: You can run `jive status -v` to see all of the services and whether they're enabled or disabled.

```
jive stop
```

7. Review [Backup and Storage Considerations](#) for more on what you should be backing up. If you are also upgrading the database, see [Upgrading a Database](#).
8. From the command line, access the target host as root.

```
joe@joesbox ~ $ ssh root@targetsystem
root@targethost's password:
Last login: Mon Feb 14 14:00:56 2011 from joesbox.example.com
```

9. If you haven't already copied the Jive application RPM to each server and application node in your Jive environment, then you should do it now.

Here's an example using the Linux `scp` command to copy the package from a computer named "joesbox" to a target system at "targetsystem":

```
scp -v joe@joesbox:/Users/joe/jive.rpm root@targetsystem:/root
```

10. Upgrade the Jive application RPM on all nodes. If you have multiple cache servers in your configuration, you need to upgrade all cache servers in the cluster.

```
rpm -Uvh jive.rpm
```



Note: All elements of the application will be upgraded, including cache services and, if you have it, document conversion services. The following shows console output for a successful upgrade using the preceding command.

```
[root@targethost ~]# rpm -Uvh jive.rpm
Preparing... #####
[100%]
Stopping httpd: running stop command..stopped
Stopping webapp: running stop command.....sending SIGTERM.stopped
Stopping search: sending SIGTERM..stopped
Stopping docconverter: sending SIGTERM.....stopped
Stopping cache: sending SIGTERM...stopped
Stopping eae: sending SIGTERM.stopped
Writing upgrade version.
Wrote upgrade version.
Pre-install tasks complete.
  1:jive_sbs #####
[100%]
Removing unused files...
...done.
Fixing permissions..
...done.
Fixing symlinks...
...done.
Installing init script and logrotate...
...done.
*****
*** Post-install tasks complete! To configure the Jive platform,
*** become the jive user and run 'jive setup'.
*****
```

- 11.If the RPM output shows that any `.rpmnew` or `.rpmsave` files were written during the upgrade, merge those changes as needed.
- 12.Run `jive setup` as the `jive` user on all nodes. For more on this, see [Using jive setup](#). Do this so the Jive CLI installer can detect any service-specific startup properties that need to be set:

```
jive setup
```

- 13.Once you get the `setup ok` message on all nodes, you can start the enabled services on all nodes **except the web application nodes**. For example, start services on the Cache nodes first, and then on the Activity Engine, Search, Ingress Replicator, and Document Conversion nodes.

```
jive start
```

- 14.Run `jive start` on one of the Web Application nodes as the Jive installation user.
- 15.With a supported web browser, navigate to `http://hostname:8080`, where `hostname` is the DNS-resolvable name of the server of the Web Application node you're using to upgrade. Once there, proceed with the upgrade using the Admin Console.
- 16.After you've finished the upgrade, log into the Web Application node you used for the upgrade tasks and restart the application.

```
jive restart
```


17. Start the `webapp` and `httpd` services on all other `webapp` nodes.

```
jive start
```

18. You're finished upgrading. You can now re-install your plugins.

19. If you had any custom themes and templates, see [Upgrading Custom Themes and Templates](#).

Upgrading Custom Themes and Templates

For each customized theme and/or template, you will run a 3-way diff against core (current core version on the left, new version on the right, customized version in the middle).

1. Upgrade your existing Jive instance using the instructions in [Upgrading a Linux Package](#).
2. Copy your production themes to the target server. Your themes are located in `/usr/local/jive/applications/sbs/home/themes`.
3. Copy your production templates to the target server. Your templates are located in `/usr/local/jive/applications/sbs/home/templates`.
4. For each customized theme and template, run a 3-way diff against core (e.g., current core version on the left, new version on the right, customized version in the middle). DiffMerge is a useful tool for performing diffs.
5. Merge from the right all differences between the old and new core themes/templates. Keep your custom modifications.