Installing

Contents

Installing Jive

Learn how to install the pieces that make up the Jive platform.

You can use the following instructions to prepare for and complete an on-premise Jive installation. Jive production installations require multiple servers, a database, and expert knowledge of your enterprise systems, so read the requirements carefully before proceeding. Keep in mind that these instructions are a template and your requirements will be specific to your community. Don't forget that you should also have a plan to back up your installation!

Note: If you just want to do a quick install for evaluation or development purposes, go to Quick Installation.

If you are installing any of Jive's add-on modules, such as Jive for Outlook or Jive Mobile, check the documentation for their separate system and installation requirements.

What's New in the Installation?

The new Jive command-line interface (CLI) installer provides a better way to start, stop and configure an installation, and gives you more flexibility.

We've streamlined the command line procedure to provide a more centralized interface for running, monitoring and troubleshooting your Jive installation. The installation also now provides interactive feedback that will help you understand whether you're meeting installation and upgrade requirements and what your next steps should be. It even records your startup modifications so you can be aware of your customizations during future upgrades.

Benefits of the New Jive CLI Installer and Interface

Saves time tracking down configurations	The site-specific configurations you've always had to make at the command line when installing the Jive platform are now in one place so you can easily view and override them.
Provides tools for managing the platform	We provide a single command-line interface for starting/stopping services and configuring the startup environment.
Enables separation of roles in the IT environment	If your organization prefers to separate the root administration and application administration roles, you can now install the application in a custom location other than the default `/usr/local/jive`, and run commands using a different user account than the default "jive" user. Whether you

use this capability or not, the root user is no longer necessary for the routine operation of the platform.

Startup Property Reference

The following tables list the startup properties available for your Jive installation, grouped by the service they support.
To understand how to set these properties, see Startup Properties Commands.

Main Startup Property Name	Description
main.gc_log_file_size	The maximum size of each java gc log file (MB) before rotation. Default: 1024
main.java_bin	Path to the java binary. Default: {{main.java_home}}/bin/java
main.java_home	Directory where java is installed. Default: {{main.jive_home}}/java
main.jive_group	Primary group for the user account that runs the Jive platform. Default: jive
main.jive_home	Directory where Jive is installed. Default: /usr/local/jive
main.jive_user	User account for running the Jive platform. Default: jive
main.log_dir	Directory for all log files. Default: {{main.jive_home}}/var/logs
main.number_of_gc_log_files	The number of java gc log file each service should retain before overwriting. Default: 2
main.pidfile_dir	Directory where all process ID files are stored. Default: {{main.jive_home}}/var/run
main.skip_setup	Always skip setup warnings when starting services. Default: False
main.start_wait_seconds	Number of seconds to wait after startup before verifying that the process is still running. Default: 5
main.system_info_cmd	Command for capturing overall system info when running 'jive snap'. Default: /usr/bin/top -n1 -b -H
main.tomcat_home	Directory where tomcat is installed. Default: {{main.jive_home}}/tomcat

Cache Startup Property Name	Description
cache.admin_max_threads	Default: 60
cache.admin_port	Port on which cache service is run. Default: 6667
cache.custom_jvm_args	Used for additional custom java arguments for cache.
cache.enabled	Always start cache when 'jive start' is run without arguments. Default: False
cache.hostnames	The short-form, unqualified hostname of the machine on which the cache service is running. For configurations with multiple cache nodes, can be a comma-separated list of cache node hostnames. In this latter case, the value for cache.hostnames must be the same on each cache node (for example, same order).
cache.jmxremote_port	JMX port for the cache service. Default: 6650
cache.jvm_heap_max	Java -Xmx value for cache, in MB. Default: 2048
cache.jvm_heap_min	Java -Xms value for cache, in MB. Default: 2048
cache.max_threads	Default: 100
cache.priority	Startup priority order for cache with respect to other jive services on this machine. Default: 0
cache.snap_cmd	Command for capturing performance information from cache when running 'jive snap'. Default: {{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
cache.socket_port	Port on which cache service is run. Default: 6666
cache.stdout	File to send cache's STDOUT stream. Default: {{main.log_dir}}/cache-service.out

Docconverter Startup Property Name	Description
docconverter.custom_jvm_args	Any additional custom java arguments for docconverter go here.
docconverter.enabled	Always start docconverter when 'jive start' is run without arguments. Default: False
docconverter.http_addr	IP for docconverter to listen on. Default: 0.0.0.0

Docconverter Startup Property Name	Description
docconverter.http_monitor_addr	Monitor IP for docconverter Default: 127.0.0.1
docconverter.http_monitor_port	Monitor port for docconverter Default: 19005
docconverter.http_port	HTTP port for docconverter. Default: 19003
docconverter.imager_path	Path to the imager utility. Default: {{main.jive_home}}/bin/imager/bin/imager
docconverter.jvm_heap_max	Java -Xmx value for docconverter, in MB. Default: 256
docconverter.jvm_heap_min	Java -Xms value for docconverter, in MB. Default: 256
docconverter.jvm_heap_perm (DEPRECATED)	Java -XX:MaxPermSize value for docconverter, in MB. Default: 64
docconverter.kill_one_oo_path	Path to the script which allows docconverter to stop its running instances of openoffice. Default: {{docconverter.service_home}}/bin/kill_one_oo.sh
docconverter.openoffice_path	Path to the openoffice binary. Default: {{main.jive_home}}/opt/libreoffice4.1/program/soffice.bin
docconverter.pdf2swf_path	Path to the pdf2swf utility. Default: {{main.jive_home}}/bin/pdf2swf
docconverter.priority	Startup priority order for docconverter with respect to other jive services on this machine. Default: 0
docconverter.server_port	Server port for docconverter. Default: 19004
docconverter.service_home	Base directory for the docconverter service. Default: {{main.jive_services_home}}/docconverter
docconverter.snap_cmd	Command for capturing performance information from docconverter when running 'jive snap'. Default: {{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
docconverter.stdout	File to send docconverter's STDOUT stream. Default: {{main.log_dir}}/docconverter.out
docconverter.textextract_path	Path to the textextract utility. Default: {{docconverter.service_home}}/utils/bin/text_extract

EAE/Activity Engine Startup Property Name	Description
eae.custom_classpath_additions	Custom paths for eae's CLASSPATH environment variable can be included here (for example, Oracle database driver install paths). Colon-separated list.
eae.custom_jvm_args	Any additional custom java arguments for eae go here.
eae.custom_ld_library_path_additions	Custom paths for eae's LD_LIBRARY_PATH environment variable can be included here (for example, Oracle database driver install paths). Colon- separated list.
eae.enabled	Always start eae when 'jive start' is run without arguments. Default: False
eae.jmx_port	Port on which JMX is run. Default: 8026
eae.jvm_heap_max	Java -Xmx value for eae, in MB. Default: 2048
eae.jvm_heap_min	Java -Xms value for eae, in MB. Default: 2048
eae.priority	Startup priority order for eae with respect to other jive services on this machine. Default: 0
eae.service_home	Base directory for the Activity Engine service. Default: {{main.jive_services_home}}/eae-service
eae.snap_cmd	Command for capturing performance information from eae when running 'jive snap'. Default: {{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
eae.stdout	File to send eae's STDOUT stream. Default: {{main.log_dir}}/eae-service.out

Httpd Startup Property Name	Description
httpd.apachectl_cmd	The base command by which apachectl is invoked for httpd. Default: {{main.jive_home}}/httpd/bin/apachectl
httpd.apachectl_lock_file	Path to httpd's apachectl lock file. Default: {{main.pidfile_dir}}/jive-httpd-accept.lock
httpd.apachectl_pid_file	Path to httpd's apachectl pid file. Default: {{main.pidfile_dir}}/jive-httpd.pid

Httpd Startup Property Name	**Description**
httpd.conf	Path to the main httpd config file. Default: {{main.jive_home}}/etc/httpd/conf/httpd.conf
httpd.enabled	Always start httpd service when 'jive start' is run without arguments. Default: False
httpd.lang	Locale for httpd (HTTPD_LANG). Default: C
	httpd.port Port on which httpd listens for normal HTTP connections. Default: 8080
httpd.priority	Startup priority order for httpd with respect to other jive services on this machine. Default: 2
httpd.procname	The name under which a running httpd service will appear in the process list. Default: {{main.jive_home}}/httpd/bin/jive-httpd
httpd.server_admin_email	Admin email for httpd. Default: support@jivesoftware.com
httpd.ssl_certificate_file	Path to SSL cert when httpd.ssl_enabled=True.
httpd.ssl_certificate_key_file	Path to SSL key when httpd.ssl_enabled=True.
httpd.ssl_enabled	Enable SSL for httpd. Default: False
httpd.ssl_port	Port on which httpd accepts ssl-enabled connections when httpd.ssl_enabled=True. Default: 8443
httpd.stdout	File to send httpd's STDOUT stream. Default: {{main.log_dir}}/jive-httpd.out

Ingress-Replicator Startup Property Name	**Description**
ingress-replicator.custom_jvm_args	Any additional custom java arguments for ingress-replicator go here.
ingress-replicator.debug_port	Search debug port. Default: 29001
ingress-replicator.enabled	Always start ingress-replicator service when 'jive start' is run without arguments. Default: False
ingress-replicator.jmx_port	Search JMX port. Default: 29002
ingress-replicator.jvm_heap_max	Java -Xmx value for ingress-replicator, in MB. Default: 2048

Ingress-Replicator Startup Property Name	Description
ingress-replicator.jvm_heap_min	Java -Xms value for ingress-replicator, in MB. Default: 2048
ingress-replicator.on_out_of_memory_error	Command to run on out-of-memory error. Default: echo "Out of memory crash!"
	ingress-replicator.priority Startup priority order for ingress-replicator with respect to other jive services on this machine. Default: 0
ingress-replicator.service_home	Base directory for the ingress-replicator service. Default: {{main.jive_home}}/services/ingress-replicator-service
ingress-replicator.snap_cmd	Command for capturing performance information from ingress-replicator when running 'jive snap'. Default: {{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
ingress-replicator.stdout	File to send ingress-replicator's STDOUT stream. Default: {{main.log_dir}}/ingress-replicator/stdout.log

Search Startup Property Name	Description
search.custom_jvm_args	Any additional custom java arguments for search go here.
search.debug_port	Search debug port. Default: 27001
search.enabled	Always start search service when 'jive start' is run without arguments. Default: False
search.jmx_port	Search JMX port. Default: 27002
search.jvm_heap_max	Java -Xmx value for search, in MB. Default: 2048
search.jvm_heap_min	Java -Xms value for search, in MB. Default: 2048
search.on_out_of_memory_error	Command to run on out-of-memory error. Default: echo "Out of memory crash!"
search.priority	Startup priority order for search with respect to other jive services on this machine. Default: 0
search.service_home	Base directory for the search service. Default: {{main.jive_home}}/services/search-service

Search Startup Property Name	Description
search.snap_cmd	Command for capturing performance information from search when running 'jive snap'. Default: {{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
search.stdout	File to send search's STDOUT stream. Default: {{main.log_dir}}/search/stdout.log

Webapp Startup Property Name	Description
webapp.app_cluster_jvmroute	A string that will be added to the JSESSIONID cookie served by the webapp, to allow cookie-based load balancing.
webapp.appname	Webapp instance name. Default: sbs
webapp.context	URL context for the webapp. For example, if your site is http://my.company.com/community, "community" would be the context. Default is blank.
webapp.custom_classpath_additions	Custom paths for webapp's CLASSPATH environment variable can be included here (for example, Oracle database driver install paths). Colon- separated list.
webapp.custom_jvm_args	Any additional custom java arguments for the webapp go here.
webapp.custom_ld_library_path_additions	Custom paths for webapp's LD_LIBRARY_PATH environment variable can be included here (for example, Oracle database driver install paths). Colon- separated list.
webapp.enabled	Always start webapp service when 'jive start' is run without arguments. Default: False
webapp.http_addr	IP for the webapp's main Tomcat connector. This is set to localhost because the jive httpd service must always serve as a proxy. Default: 127.0.0.1
webapp.http_maxthreads	The maxThreads attribute for the webapp's main Tomcat connector. Default: 15
webapp.http_monitor_addr	IP for the webapp's monitoring Tomcat connector. Default: 127.0.0.1

Webapp Startup Property Name	Description
webapp.http_monitor_maxthreads	The maxThreads attribute for the webapp's monitoring Tomcat connector. Default: 15
webapp.http_monitor_port	Port for the webapp's monitoring Tomcat connector. Default: 9002
webapp.http_monitor_proxy_name	The webapp's monitoring Tomcat connector's proxyName attribute.
webapp.http_monitor_proxy_port	The webapp's monitoring Tomcat connector's proxyPort attribute. Default: {{webapp.http_monitor_port}}
webapp.http_monitor_proxy_scheme	The webapp's monitoring Tomcat connector's scheme attribute. Default: http
webapp.http_port	Port for the webapp's main Tomcat connector. Default: 9001
webapp.http_proxy_name	The webapp's main Tomcat connector's proxyName attribute. This is your external load balancer's address.
webapp.http_proxy_port	The webapp's main Tomcat connector's proxyPort attribute. This is your external load balancer's port. Default: {{httpd.port}}
webapp.http_proxy_scheme	The webapp's main Tomcat connector's scheme attribute, http or https. Choose based on the type of connection provided by your external load balancer. Default: http
webapp.jive_application	Directory containing the webapp application files. Default: {{webapp.jive_base}}/application
webapp.jive_base	Base directory for the web application. Default: {{main.jive_home}}/applications/ {{webapp.appname}}
webapp.jive_instance_home	Webapp home directory. Default: {{webapp.jive_base}}/home
webapp.jmx_enabled	Enable JMX on the webapp. Default: False
webapp.jmx_port	Port on which to run JMX, if webapp.jmx_enabled=True. Default: 6651
webapp.jvm_heap_max	Java -Xmx value for webapp, in MB. Default: 2048

Webapp Startup Property Name	Description
webapp.jvm_heap_min	Java -Xms value for webapp, in MB. Default: 2048
webapp.jvm_heap_perm (DEPRECATED)	Java -XX:MaxPermSize value for webapp, in MB. Default: 512
webapp.log4j_conf	The path to a minimal log4j.properties file used prior to initializing the application. Default: {{webapp.jive_base}}/conf/log4j.properties
webapp.priority	Startup priority for webapp with respect to other jive services on this machine. Default: 1
webapp.server_port	Webapp server port. Default: 9000
webapp.snap_cmd	Command for capturing performance information from webapp when running 'jive snap'. Default: {{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
webapp.stdout	File to send webapp's STDOUT stream. Default: {{main.log_dir}}/{{webapp.appname}}.out
webapp.stop_cmd_jvm_heap_max	Java -Xmx value for the webapp stop command, in MB. Default: 512
webapp.stop_cmd_jvm_heap_min	Java -Xms value for the webapp stop command, in MB. Default: 512
webapp.stop_cmd_jvm_heap_perm (DEPRECATED)	Java -XX:MaxPermSize value for the webapp stop command, in MB.
webapp.work_dir	Temp file directory for webapp. Default: {{main.jive_home}}/var/work/{{webapp.appname}}

Connecting Through a Proxy Server

Certain core components and features, including the Recommender, Mobile, and Apps Market services, require Jive to access information from across the firewall. If you use a proxy server to access the Internet, setting up the proxy connection using the Proxy Server settings in the Admin Console ensures you can connect to Mobile and the Apps Market. You can also set exceptions if you want to connect to certain hosts directly, rather than through the proxy server.

Fastpath: Admin Console: System > Settings > Proxy Server

See Advanced Proxy Configurations below for some situations that may require additional workarounds.

Setting JVM Properties for Recommender

Connecting to the Recommender service requires setting several JVM properties after you complete the Activity Engine installation. To set them:

1. Use `jive set eae.custom_jvm_args` to add the required java args to the service invocation:

   ```
   jive set eae.custom_jvm_args -Dhttp.proxyHost=webproxy_address -
   Dhttp.proxyPort=webproxy_port -Dhttp.proxyUser=webproxy_user  -
   Dhttp.proxyPassword=webproxy_pwd
   ```

   For example, add:

   ```
   -Dhttp.proxyHost=webproxy.eng.jiveland.com -Dhttp.proxyPort=3128 -
   Dhttp.proxyUser=jive -Dhttp.proxyPassword=jive
   ```

   The proxyUser and proxyPassword properties are required. If your proxy server does not require a user name and password, include these options with blank values.

   

   Warning: If you have already set eae.custom_jvm_args, you need to append the existing setting to the string above, or it will be overwritten.
2. If the system is NOT the production instance go to **System** > **Management** > **System Properties** and set the system property `jive.eae.instance.type` to the value 2.
3. Restart the Activity Engine instance and the application servers.

Advanced Proxy Configurations

If you have DNS proxying enabled, you also need to set the `jive.apps.proxy.whitelist.cidrs` system property to include the IP address (single node) or CIDR addresses of your proxy server. Domain names are not supported. You can separate multiple addresses with spaces.

If your site uses client certificate validation, you should contact Support for assistance.

Reverse proxying can modify data in ways that are not compatible with running Jive. Reverse proxies are often configured to reject GETs with special characters in the query string, strip the bodies from PUT commands, and add prefixes to cookie names. For example, Jive Apps require the colon (:) character to be supported in URLs. If you use IIS-based reverse proxying with .NET 4.5 or earlier, colons are blocked by default. In this case, upgrading to .NET 5.0 resolves the problem.

Preparing to Connect to Jive-Hosted Services

To ensure a successful installation, you need to make sure you can connect successfully to several component services that are hosted by Jive.

In addition to the core components of Jive that you install on your own servers, running Jive requires connections through your firewall to the Apps Market service, the Recommender service, and, if you use them, the Mobile Gateway, Search and Video services. The following sections provide the ports and addresses you'll use when ensuring your firewall can access the correct ports and domains for these services. They also provide some pointers for successful setup.

General Best Practices

For all these components, you need to ensure that your proxy server is configured to access resources outside the firewall. See Connecting Through a Proxy Server for more information.

Recommender Service

Make sure your Activity Engine servers can connect through the firewall using the following settings.

Component	Jive CLI Name	Port(s)	Direction	Domain(s) or IPs
Activity Engine	eae	TCP port: 7020 JMX port: 7021,8026 RMI ports: 33030,56844	Open	

After installation is complete, you can check the status each Activity Engine's connection to the Recommender service using the Activity Engine page in the Admin Console. Navigate to **System** > **Settings** > **Activity Engine** and check the Recommender column of the Activity Engine Overview table.

Note that if you relocate an instance by changing the jiveURL, and you then enable and disable the Recommender on that instance, the Recommender restarts with a new ID for the instance. Recommendations from before the jiveURL changed will be lost.

Search Service

With an on-premise installation, you can choose from either Jive Cloud search or On-Premise Search. The Jive Cloud Search offers socially contextual search, and requires a simple support case to whitelist your IP Addresses. Jive Cloud and Hosted both use Cloud Search. For more about the benefits of Jive Cloud Search, see About Cloud Search.

On-Premise Search uses the On-prem search service installed in your Jive environment.

The following table shows which ports and domains are required so the web application nodes can contact the Search service:

Component	Jive CLI Name	Port(s)	Direction	Domain(s) or IPs
Cloud Search Service		443	Outbound	directory-service.phx1.jivehosted.com or directory-service.ams1.jivehosted.com

Component	Jive CLI Name	Port(s)	Direction	Domain(s) or IPs
On-Premise Search Service	search-service	Service port: 30000 Debug port: 27001 JMX port: 27002	Open	localhost

You can check the status and connection to the Search service using the Search page in the Admin Console. Navigate to **System** > **Settings** > **Search** You can also use the Admin Console's Search page to make changes to the location or type of Search service. If you change the type of search service (Cloud Search or On-Premise Search), you need to perform a full index rebuild.

Apps Market Service

Make sure the following ports and domains are enabled so the web application nodes can contact the Market.

Component	Jive CLI Name	Port(s)	Direction	Domain(s) or IPs
Jive Apps Market		80 or 443	Outbound	market.apps.jivesoftware.co gateway.jivesoftware.com, apphosting.jivesoftware.com developers.jivesoftware.com

After you complete your Jive installation, make sure you have enabled the legacy REST services under **System** > **Settings** > **Web Services** > **Legacy Web Services** and the Core API under **System** > **Settings** > **Web Services** > **Core API**. Complete instructions for setting up and troubleshooting your connection to the Apps Market service can be found under Setting Up Apps.

Note that if you relocate an instance by changing the jiveURL, and you then enable and disable the Apps Market on the instance, the Market restarts with a new ID for the instance. Installed Apps from before the jiveURL changed may be lost or work incorrectly. The correct path is to relocate the instance with the Market enabled, so that the ID remains the same.

Mobile Push Service (optional)

If you plan to have your community members receive Jive push notifications, you'll need access to the Jive Mobile Push Service. Make sure the following ports and IPs are enabled so the web application nodes can contact it.

Component	Port(s)	Direction	Domain(s) or IPs
Mobile (all locations, including EMEA) when you are sending push notifications to the publicly available apps.	443	Outbound from Jive instance	mobilepush.prod.jiveon.com (204.93.64.255 and 204.93.64.252)
Mobile (all locations, including EMEA) when you are using your custom branded iOS app with your own push notification certificate	• TCP port 5223 (used by devices to communicate to the APNs servers) • TCP port 2195 (used to send notifications to the APNs) • TCP port 2196 (used by the APNs feedback service) • TCP Port 443 (used as a fallback on Wi-fi only, when devices are unable to communicate to APNs on port 5223)	Outbound from Jive instance	
Mobile (all locations, including EMEA) when you are using your custom branded Android app with your own Google Cloud Messaging key	443	Outbound from Jive instance	https://android.googleapis.com/gcm/send

Jive Present

Jive Present provides specified community users access to the Jive Present for iPad application.

Component	Port(s)	Direction	Domain(s) or IPs
Jive Present	443	Inbound to Jive instance	188.93.102.111 188.93.102.112 188.93.102.115 188.93.103.35 188.93.103.36 188.93.103.37

Video Service

Setting up the firewall for video is complex. For complete information, see the Jive Video Help listed separately on our main documentation page.

Managing Client Certificates

In the rare case where your corporate network uses client certificates for authentication, you will need to configure Jive so it can authenticate properly. You can use the Admin Console to choose how to handle client certificates. If you choose a method that allows for storing certificates in Jive, you can also view and manage them in the Admin Console. Jive interface accepts PKCS12 encrypted keys, just as your browser does. Uploading a new certificate and key takes effect immediately, and all the nodes in a cluster share the same keystore.

Fastpath: Admin Console: System > Settings > Client Certificates

Certificate Management Strategies

- Select **Java** as the client certificate strategy if you're planning to use Java's keytool to manage certificates outside Jive using Java system properties. The Client Certificates dialog then becomes read-only, and you can't upload any certificates. More information about the Java keytool is available at the Oracle website.
- Select **Issuer** as the client certificate strategy if you want to select certificates by issuer. You'll need to provide the key and the certificate password when you upload your client certificates to the Jive keystore.
- Select **Domain** as the client certificate strategy if you want to specify the certificates to use for a specific domain. You'll need to provide the domain and the certificate password used to access the cert file when you upload your client certificates to the Jive keystore.

Testing Certificate Validation

If you put a test URL in the box and click **Test Connectivity**, you'll get a detailed report on client and server connectivity.

Installation Overview

When you install the Jive platform, you'll perform several steps in a specific sequence. Here is an overview of all the installation steps and the order in which you should perform them:

	Required or Optional	Install Instructions
1	Required	Review the following topics to understand a typical configuration, required components and sizes, and how to scale your system: • System Architecture drawing • System Requirements • RPM Dependencies by Operating System • Pre-Installation Requirements as Root on CLI • Network Requirements
2	Required	Review the information in Connecting to Jive-Hosted Services to ensure your servers can communicate with the externally hosted components. If you use a proxy server, make sure it's configured correctly to access the required external resources. See Connecting Through a Proxy Server.
3	Required Activity Engine Database, Core Databases, Analytics Database	Install the Activity Engine, Core, and Analytics databases each on their own server as described in Database Configuration and Best Practices, and you have created users on them.
4	Required Activity Engine	Install the Jive Linux package on your Activity Engine node. This node should be able to access the Recommender Service as described in Connecting to Jive-Hosted Services.
5	Required for On-Premise Search only. For information on Jive Cloud Search, see About Cloud Search.	If you are using On-Premise Search, install the Jive Linux package on your Search node. Make sure the web app node can access the Search service as described in Connecting to Jive-Hosted Services.
6	Required Web Apps	Install the Jive Linux package on your web app nodes. Any web app nodes will need access to the domains for Apps Market and Mobile Gateway, directly or through your proxy server, as described in Connecting to Jive-Hosted Services.
7	Required Configured Cache Server	Install the Jive Linux package on the cache server(s). See the System Architecture drawing to better understand a typical configuration and how to scale your system.

Required or Optional	Install Instructions
Required	With a supported web browser, navigate to http://<hostname>/, where hostname is the DNS-resolvable name of the server where you installed the Jive application on your primary web application node. There, you will be prompted to finish configuring the Jive application via the Admin Console setup wizard.
Required	In the Other Settings page of the setup wizard, you'll be asked to choose between the Cloud and On-Premise Search service. If you're using On-Premise Search, make sure you select **On-Premise Search**. If you're using Cloud Search, you'll need to contact support to get your IP addresses whitelisted. For more on Cloud Search, see About Cloud Search.
Required	Configure your cluster. Configurations require a clustered environment, unless you are only evaluating Jive. To configure a clustered environment, you need to install the Jive Linux package on each node of the cluster, and configure it appropriately.
Required	Install the Jive Linux package on the document conversion node. This module allows documents to be previewed in Jive without modifying their original format. For more information, and to complete the required server configuration for document conversion, see Setting Up Document Conversion. **Note:** Because you'll need to use the Admin Console to create the connection, you should finish the installation and configuration steps in this checklist before you complete the document conversion node setup.
Required	With a supported web browser, navigate to http://<hostname>/, where hostname is the DNS-resolvable name of the server where you installed the Jive application on your primary web application node. Login as the system administrator and use the instructions for setting up a binary storage provider.
Optional	If you plan to populate your community with users synchronized from your LDAP or Active Directory implementation, see the LDAP and Active Directory Guide.
Required	Use the post-installation tasks to perform some one-time configurations for Jive, such as configuring the location for text extraction for Search.

CLI Installation Tasks

Installation tasks include installing the RPM, using `jive setup`, and copying the configuration to other nodes in clustered environments.

Installing the Jive Package and Starting Up

Jive's new installer, the command-line interface (CLI), provides a time-saving way to track down configurations and manage the platform.

Jive is compatible with a number of hardware configurations as well as network topologies. To understand the recommended deployment configuration for an on-premise installation, see Jive Enterprise Architecture.

What You'll Need

To install Jive using the RPM, you'll need the following:

- At least six servers that meet the minimum specified hardware requirements described in System Requirements.
- These servers should be running Linux. See the System Requirements for supported versions.
- A bash shell to run the install commands.
- SSH access to the servers so you can copy the RPM there for installation.
- Ability to become the root user on the servers where the installation is performed, commonly via SSH, or less commonly through user interface access such as VNC.

  Note: If you want to run a non-standard installation, you could separate installation tasks into root user and non-root user tasks. You'd need a root user to complete the Pre-Installation Requirements as Root on CLI, and this root user would have to create a non-root user who can run the installation. For more on how to create a non-root user see Installing Jive with Non-Root User.
- Make sure you have installed the packages described in RPM Dependencies by Operating System and you complete the Pre-Installation Requirements as Root on CLI. They include copying the Jive and `pdswf` RPMs to the servers and modifying a few configuration files.

Install the Package on All Nodes

You will need to install the Jive Linux package on the following nodes. You can install the RPM on these nodes in any order, but use the Installation Overview for our recommended order.

- Activity Engine node
- Search node
- Web Application nodes (2 or more, where node 1 is the primary Web Application node)
- Cache server
- Document Conversion node

Installation Steps Using the Jive CLI

The following installation steps are the most common approach to installing the Jive platform:

1. From the command line, access the target host as root. For example, the following illustrates using the `ssh` command to access the server at targethost as the root user.

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

2. 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
```

3. Install the Jive application RPM on all nodes using an `rpm` command such as the following. The `U`, `h`, and `v` options are provided to indicate install/upgrade with hash indicators, and to be verbose during the installation. (Your copy of the Jive RPM file -- here, `jive.rpm` -- will have a slightly different name.)

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

4. When the installation finishes, the Jive CLI installer asks you to become a jive user and run `jive setup`, which you can do by using the following command. For more on this, see Using jive setup on page 22.

```
su - jive
jive setup
```

5. Address any configuration issues and suggestions that `jive setup` provides. Re-run `jive setup` until you get the `setup ok` message.
6. Set up your Core application, Activity Engine, and Analytics (optional) databases using Setting Up New Databases. If you'll be using a database whose driver is not included, ensure its driver is in the application's classpath by following the steps in Database Prerequisites.
7. After you have installed the Jive package and run `jive setup` on all of your nodes, you can enable the services as described in the next section.

Enabling the Services

1. Enable the services on all of the nodes as a jive user.

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

On this node	Run this command as jive user
Activity Engine	**`jive enable eae`**
Search	**`jive enable search`**

On this node	Run this command as jive user
Web application nodes	`jive enable webapp` `jive enable httpd`
Cache server(s) and cluster nodes	`jive enable cache`
Document Conversion	`jive enable docconverter`

2. After enabling the correct service(s), re-run `jive setup`. For more on this, see Using jive setup on page 22. Do this so the Jive CLI installer can detect any service-specific startup properties that need to be set:

```
jive setup
```

3. Once you get the `setup ok` message on your node, you can start the enabled services on Jive.

```
jive start
```

4. With a supported web browser, navigate to http://<hostname>/, where hostname is the DNS-resolvable name of the server where you installed the Jive application on your primary web application node. There, you will be prompted to finish configuring the Jive application via the Admin Console setup wizard. If you plan to populate your community with users synchronized from your LDAP implementation, the setup screens are included in this wizard.
5. See Post-Installation Tasks for your next steps.

Using `jive setup`

Use `jive setup` after you install the RPM. It provides system requirements and suggestions for configuring your Jive instance.

As you go through the `jive setup`, it will provide suggestions for your configuration. Use the command-line output to update your configurations. After you make changes, run `jive setup` until you get the `setup ok` message.

Note: Running `jive start` automatically runs `jive setup` before starting jive unless you disable it as shown in the following section.

Disabling `jive setup`

You can skip jive setup by doing one of the following:

- Tell the start process to not use setup, and only start Jive:

```
jive start --skip-setup
```

- Set the main.skip_setup startup property to true:

```
jive set main.skip_setup true
```

Copying Configurations to Other Nodes

After running `jive setup`, and configuring all of the overrides, you can move the overrides configuration file to another node that requires the same setup.

After going through `Jive setup`, the Jive CLI updates the `jive.properties` file, which is where it saves overrides to the startup properties. You can copy the `jive.properties` file to another node that needs the same setup. For example, after configuring your first web application node, you can copy the *`Jive Installation Directory`*`/etc/jive.properties` file to the second node. You also need to manually copy the encryption keys in the `/usr/local/jive/applications/`*`app_name`*`/home/crypto` directory to the new node. After that, you can just run `jive start` to start the node.

Troubleshooting the Installation

The Jive installation uses Linux RPM, a widely tested and used application that is very unlikely to fail. However, if you run into trouble during an installation, you can delete and start over as described here.

Note: You'll find the installation log files on the target computer at /usr/local/jive/var/logs.

Unsatisfied Dependencies

The Jive application RPM depends on the presence of several low-level system packages that are common to nearly all configurations of Jive's supported Linux distributions. Also, the Jive application RPM depends on three high-level packages. If any of these packages (system or high-level) is not present, the RPM subsystem will warn you, then refuse to install. When you see these warnings, simply install the missing packages using RPM, then install Jive as described in the instructions.

Unsatisfied dependencies appear as an error when attempting to install the Jive application:

```
[root@targethost ~]# ls -l
total 202068
-rw-r--r-- 1 root root 206701420 Jan 20 16:03 jive_sbs-5.0.0-78310.i386.rpm
-rwxr-xr-x 1 root root      1347 Oct  7 16:14 updateDNS.sh
[root@targethost ~]# rpm -ivh jive_sbs-5.0.0-78310.i386.rpm
error: Failed dependencies:
     bash >= 3.2 is needed by jive_sbs-5.0.0-78310.i386
     sysstat >= 7 is needed by jive_sbs-5.0.0-78310.i386
```

Depending on the host configuration, it may be possible to install the dependencies directly using system tools. For example, in RedHat Enterprise Linux, the "yum" command can install dependencies via network repositories. The following demonstrates how to install the dependencies shown in the error above.

```
[root@targethost ~]# yum install bash-3.2 sysstat
Loading "installonly" plugin
Setting up Install Process
Setting up repositories
extras                    100% |=========================| 1.1 kB    00:00
updates                                                      951 B
 00:00
base                      100% |=========================| 1.1 kB    00:00
addons                    100% |=========================|  951 B    00:00
Reading repository metadata in from local files
```

```
primary.xml.gz            100% |=========================|  90 kB    00:00
########################################################## 295/295
primary.xml.gz                                               369 kB
 00:03
########################################################## 796/796
primary.xml.gz            100% |=========================| 853 kB    00:01
########################################################## 2458/2458
Parsing package install arguments
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for sysstat to pack into transaction set.
sysstat-7.0.2-1.el5.i386. 100% |=========================|  15 kB    00:00
---> Package sysstat.i386 0:7.0.2-1.el5 set to be updated
---> Downloading header for bash to pack into transaction set.
bash-3.2-21.el5.i386.rpm  100% |=========================|  55 kB    00:00
---> Package bash.i386 0:3.2-21.el5 set to be updated
--> Running transaction check
Dependencies Resolved

================================================================================
 Package                 Arch       Version          Repository        Size
================================================================================
Installing:
 sysstat                 i386       7.0.2-1.el5      base              168 k
Updating:
 bash                    i386       3.2-21.el5       base              1.9 M
Transaction Summary
================================================================================
Install      1 Package(s)
Update       1 Package(s)
Remove       0 Package(s)
Total download size: 2.0 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): sysstat-7.0.2-1.el 100% |=========================| 168 kB    00:00
(2/2): bash-3.2-21.el5.i3 100% |=========================| 1.9 MB    00:02
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating  : bash                         ######################### [1/3]
  Installing: sysstat                      ######################### [2/3]
  Cleanup   : bash                         ######################### [3/3]
Installed: sysstat.i386 0:7.0.2-1.el5
Updated: bash.i386 0:3.2-21.el5
Complete!
```

After dependencies have been resolved, the package should install normally.

Insufficient System Memory

The Jive platform requires a minimum of 3GB of RAM to operate effectively for an enterprise environment. If sufficient memory is not available on the target installation system, the installer will provide a warning at installation time similar to the example below.

```
[root@targethost ~]# rpm -ivh jive_sbs-5.0.0-78310.i386.rpm
Preparing...                ###########################################
 [100%]
   1:jive_sbs               ###########################################
 [100%]
Writing installation version.
```

```
Wrote installation version.
Executing Jive post-install configuration.
Creating jive group jive.
Creating jive system user jive.
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Marking all upgrades as complete.
WARNING: this host does not have sufficient RAM to run a production Jive
 system.
A minimum of 3GB is required to host the application and HTTPD servers. 4GB
 is required to
run a locally hosted database.
Starting Jive System Daemon.
Performing Jive system configurations.
Disabling CPU frequency stepping.
. . .
Jive post-install configuration complete.
```

In the above example, note the message “WARNING: this host does not have sufficient RAM to run a production system. A minimum of 3GB is required to host the application and HTTPD servers. 4GB is required to run a locally hosted database.”

Despite this warning, the package does install correctly; however, further errors are noted on the output line: "Failed to start application sbs. See log file at '/usr/local/jive/var/logs/sbs.out'." The contents of this log file indicate:

```
[root@targethost ~]# cat /usr/local/jive/var/logs/sbs.out
SCRIPT_DIR=/usr/local/jive/applications/sbs/bin
JIVE_BASE=/usr/local/jive/applications/sbs
```

```
Creating temp directory at /usr/local/jive/var/work/sbs.
Starting application sbs
Error occurred during initialization of VM
Could not reserve enough space for object heap
```

Starting Over

In the unlikely event that something goes wrong during installation and you want to start over, you can uninstall. When uninstalling, you don't specify the RPM filename, as you did when installing. Instead, provide the logical name by which the RPM now knows the application: `jive_sbs`. Here's an example using `rpm -e` for uninstalling:

```
rpm -e jive_sbs
```

If you want to be sure you've removed all remnants of the installation, delete the destination directory created by the RPM with:

```
rm -rf /usr/local/jive
```

Changing the Root Context (optional)

Jive Apache uses the context element of a web application to delegate a URL to your Jive web application. You can set the context to something other than root, if needed.

By default, Jive installs and configures using the root context. For example, `http://yourcommunity.com`. This is the recommended configuration and default installation behavior, but you can also specify a non-root context such as `http://yourcommunity.com/engage`. If you've already installed Jive, see Changing the Root Context on an Existing Installation on page 26 to change the context to something other than root.

Important: Never use "community" as your non-root context because the application already uses this string.

To change the root context on a new install:

1. Add the non-root context to the startup property by using the following command, where "engage" is an example of adding new context. Setting this property makes the Jive web application available at `http://yourcommunity.com/engage` instead of `http://yourcommunity.com`.

   ```
   jive set webapp.context engage
   ```

2. Restart the Web Application node by using the following command.

   ```
   jive restart webapp
   ```

3. Restart the Jive Apache service by using the following command.

   ```
   jive restart httpd
   ```

Changing the Root Context on an Existing Installation

Use these instructions if you've already installed Jive, and you need to change the context to something other than root.

To change the root context on an existing installation:

1. Add the non-root context to the startup property by using the following command, where "engage" is an example of adding new context. Setting this property makes the Jive web application available at `http://yourcommunity.com/engage` instead of `http://yourcommunity.com`.

   ```
   jive set webapp.context engage
   ```

2. Change the jiveURL system property to reflect the new context for the web application. To do this, go to **Admin Console: System** > **Management** > **System Properties** and click the edit icon. For more on the jiveURL, see Changing the jiveURL.
3. Restart the Web Application node by using the following command.

   ```
   jive restart webapp
   ```

4. Restart the Jive Apache service by using the following command.

   ```
   jive restart httpd
   ```

Alternate Installation Scenarios with Jive CLI

You can use the Jive CLI to customize your Jive installation so it fits your needs.

If the default method of installation doesn't work for you, choose a scenario that makes sense for your environment.

Installing Jive Without Root Access

Create a new user who performs both the RPM install and runs the platform. You need to create this user in advance.

You need to use the following procedure on each node you're installing Jive.

Note: Installing without root access, does not install an init script or a logrotate cron task. To perform these tasks, you'll need root access to run `sbin/install_init_script_and_logrotate.sh`, or equivalent.

1. Someone with root access needs to review Pre-Installation Requirements as Root on CLI and complete the required tasks.
2. Create a group for your user. We use `myjivegroup` in the following example.

```
groupadd myjivegroup
```

3. Create the user and add it to the group you created in the previous step.

```
useradd -g myjivegroup -c "My Jive Community Service Account" -d /opt/
apps/myjive -m -s /bin/bash myjiveuser
```

Note:

- Make sure this user has /bin/bash as shell.
- You must install Jive in the user's home directory. In this example, the installation directory is `/opt/apps/myjive`.

4. Initialize an RPM database as the new user in that user's home directory.

```
su - myjiveuser
```

```
mkdir rpmdb
```

```
rpm --initdb --dbpath /opt/apps/myjive/rpmdb
```

5. After downloading the Jive RPM and copying it to the node where you're installing Jive, you need to verify that the dependencies are met.

```
rpm -Vp --nofiles path/to/jive.rpm
```

6. Install the RPM as your new user.

```
JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive
 rpm --dbpath /opt/apps/myjive/rpmdb -Uvh --prefix=/opt/apps/myjive --
nodeps path/to/jive.rpm ]
```

7. For the docconverter node, install the `pdf2swf` RPM.

```
JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive
 rpm --dbpath /opt/apps/myjive/rpmdb -Uvh --prefix=/opt/apps/myjive --
replacefiles --nodeps path/to/pdf2swf.rpm
```

8. Source the .bash_profile that was just installed, or log out and back in again as your new user. For example, `myjiveuser`.
9. Run `jive setup`, which you can do by using the following command. For more on this, see Using jive setup on page 22.

```
jive setup
```

Installing in a Different Directory With a Different User

Install the RPM normally as root, but with an installation user and group other than "jive", and/or an installation directory other than "/usr/local/jive".

With root access, you can install Jive and create a non-default user to run the platform. You can skip Steps 1 on page 28 and 2 on page 28 if you don't need to create your user ahead of time.

Note: If you are not creating a user in advance, you will specify a user and group during the RPM installation using the JIVE_USER and JIVE_GROUP environment variables. These variables default to "jive" and "jive". For an example of how to set these variables, see Step 3 on page 29

To install and set up Jive:

1. Create a group for your user. We use `myjivegroup` in the following example.

```
groupadd myjivegroup
```

2. Create the user and add it to the group you created in the previous step.

```
useradd -g myjivegroup -c "My Jive Community Service Account" -d /opt/
apps/myjive -m -s /bin/bash myjiveuser
```

Note:

- Make sure this user has /bin/bash as shell.
- You must install Jive in the user's home directory. In this example, the installation directory is `/opt/apps/myjive`.

3. Log in as the root user to perform an RPM installation, so that the Jive platform will be runnable by the new user and installed in the new user's home directory In the following example, we use "myjiveuser","myjivegroup" and "/opt/apps/myjive" as the user, group and installation directory.

```
JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive rpm
 -Uvh --prefix=/opt/apps/myjive path/to/jive.rpm
```

Parameters	Description
`JIVE_USER=myjiveuser` `JIVE_GROUP=myjivegroup`	Uses these parameters to specify the non-default user that will run the platform. If you don't use these parameters, the values default to "jive" and "jive". The RPM installer creates this user and group if it does not exist.
`JIVE_HOME=/opt/apps/myjive` `--prefix=/opt/apps/myjive`	Use JIVE_HOME and the "--prefix" flag to install Jive in an alternate location. Installing the RPM without using these parameters installs in the default location, which is `/usr/local/jive`. **Note:** You can change the installation directory even if you are using the default "jive" user.

4. For the docconverter node, log in as the root user to install `pdf2swf` RPM.

```
JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive rpm
 -Uvh --prefix=/opt/apps/myjive --replacefiles path/to/pdf2swf.rpm
```

5. When the installation finishes, become the user you created during the install and run `jive setup`, which you can do by using the following command. For more on this, see Using jive setup on page 22.

```
su - myjiveuser
```

```
jive setup
```

Running Jive on Port 80 Without SSL

Although not recommended in production, you can run Jive on port 80 without using SSL if you configure webapp proxy settings and you redirect ports using iptables.

To honor a number of customer requests, we removed the requirement of being a privileged (root) user to install Jive. As a result, Jive now runs on port 8080 because Jive is now installed by non-privileged (jive) user and that user cannot use port 80.

Beginning with Jive 7, we insist you run Jive with the more secure SSL. If you want to change this, you now need to make a few Linux configurations to allow access to your site using port 80 without SSL.

To set up a Jive instance without SSL that listens on port 80:

1. Configure the webapp proxy settings using the following commands:
   - `jive set webapp.http_proxy_name <your_hostname_here>`
   - `jive set webapp.http_proxy_port 80`
   - `jive set webapp.http_proxy_scheme http`
2. Redirect ports using iptables using the following commands as root user:
   - `iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080`
   - `iptables -A OUTPUT -t nat -p tcp -d 127.0.0.1 --dport 80 -j REDIRECT --to-ports 8080`
   - `iptables -A OUTPUT -t nat -p tcp -d $(host `hostname` | awk '{print $NF}') --dport 80 -j REDIRECT --to-ports 8080`
3. If you want these settings to persist on reboot use the following commands, also as root user:
   - `mv /etc/sysconfig/iptables /etc/sysconfig/iptables-old && iptables-save > /etc/sysconfig/iptables`
   - `chkconfig iptables on`

Setting Up Document Conversion

Some documents -- including PDFs and those from Microsoft Office -- are supported in a preview view in Jive. To convert content from its native format into a form that can be previewed without altering the original document, you'll need the Document Conversion module, which you'll need to deploy on a server that is separate from your core Jive production instances.

We support converting the following file types on Office 2003 and 2007:

- doc
- ppt
- docx
- pptx
- xls
- xlsx
- pdf

Note: For information about managing conversion attempts and reconverting documents if necessary, see Managing Document Conversion.

Here is an overview of the steps you'll perform to set up Document Conversion:

1. Set up a production instance of the Jive application (see Installing the Linux Package). You'll be devoting one node in your installation to document conversion.
2. Install the Jive platform RPM on your conversion node machine. Then disable the services not related to document conversion. For more information, see Installing and Configuring on the Conversion

Machine. Download and install the correct RPM for the PDF2SWF utility on the conversion node machine. You can find the RPMs here.

Note: On the document conversion node, use the `--replacefiles` flag if you receive the following error when installing either the jive_sbs rpm or jive_pdf2swf.

```
file /usr/local/jive/bin/pdf2swf from install of rpm_name conflicts
                              with file from package other_rpm_name the flag
```

3. Enable the Document Conversion service using the following commands:

```
jive enable docconverter
```

```
jive start
```

4. On the application node, configure the application to communicate with the conversion machine(s).
5. If you want to set up secure communication to the conversion machine, see Setting Up SSL for Document Conversion.

Setting Up SSL for Document Conversion

If you have an SSL certificate, you can set up secure communication by editing the docconverter/conf/server.xml file and specifying the new secure URL in your Document Conversion Settings.

Before you can set up secure communication with your Document Conversion server, you need to acquire an SSL certificate.

1. Edit the `/usr/local/jive/services/docconverter/conf/server.xml` file and add a connector to listen on port 8443.
   `<Connector port="8443" maxThreads="200" scheme="https" secure="true" SSLEnabled="true" SSLCertificateFile="/usr/local/jive/services/docconverter/home/jive.crt" SSLCertificateKeyFile="/usr/local/jive/services/docconverter/home/jive.key" clientAuth="optional" />`

   where `SSLCertificateFile` is the certificate file and `SSLCertificateKeyFile` is the key file.

   For more information on setting up Tomcat and https, see: http://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html
2. Make sure the SSL engine is on.
   `<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on"/>`
3. Restart the document conversion service by running the following command as the jive user.
   `jive restart docconverter`
4. Go to **System** > **Settings** > **Document Conversion Settings** and edit the Conversion Service Settings to specify the new secure URL and port.
   https:// conversion-node :8443 /conversion/v1
5. Verify that you can run all conversion tests successfully.

Configuring a Document Conversion Node Connection

Use the Document Conversion Settings page to configure the node that hosts the coreJive application, so the main application knows how to communicate with any conversion machines you've set up. Before you use the settings on this page, you should already have set up a document conversion node. For an overview of Document Conversion setup, see Setting Up Document Conversion.

Fastpath: Admin Console: System > Settings > Document Conversion

The Document Conversion Settings page is pre-populated with the default values for most installations. Use the following steps to ensure your setup is correct.

To enable document conversion:

1. Select **Enabled** under Document Conversion to enable this feature.
2. If you want to stop requiring Flash for previewing uploaded documents, then select **Enabled** under HTML5 Document Previews.
3. Click **Add a Node** to start configuring a conversion machine.
4. Enter the IP address or hostname for the conversion machine.
5. Switch the way you access the conversion service URL from http to https by setting this in Conversion Service Settings field.
6. If you want to exclude some file types from conversion, enter a comma-separated list in **Disabled Extensions**. For example, if you don't want to convert Excel files, type:

```
xls, xlsx
```

Troubleshooting Document Conversion Machine Setup

If your document conversion tests are failing, try investigating the following areas to resolve the problem:

1. Check that port 19003 is open between the application node and the conversion node by executing a telnet command. For example, run `telnet 10.61.32.156 19003`.
2. Check the log for the doc-converter service for startup exceptions. You can find the log at `/usr/local/jive/var/logs/docconverter.out`.
3. If the office to pdf test is failing in the Document Conversion Setup screen, execute `jive status -v` to verify that docconverter has open ports 8820, 8821, 8822, 8823, 8824.

Setting Up a Cache Server

Use the same Linux package that you used to install the application server(s) to install a separate cache server that application servers can use in a cluster.

If your installation uses a single application node, the installation will not enable the cache services; instead, the installation will use the local cache installed with the application server. When you have a multi-node configuration, use the following steps to set up a cache service in the cluster.

To install a cache server:

1. Install the RPM as described in the (installation documentation).
2. Because the cache server machine's only function will be operating as a cache server, only enable the cache service. To do this, type the following as jive user:

   Note: If you're logged in as root, you can use `su - jive` to become the jive user.

   ```
   jive enable cache
   ```
3. Configure the cache server with its address.
4. Register and start the caching service by using the following commands. For more on setting up multiple cache servers and high-availability, see Configuring the Cache Servers for High-Availability

   ```
   jive set cache.hostnames hostname
   jive start
   ```

   The cache service writes several log files to $JIVE_HOME/var/logs/. These are:

   - cache-service.out -- Cache startup messages, output from the cache processes, showing start flags, restarts, and general errors.
   - cache-gc.log -- Output from garbage collection of the cache process.
5. If you haven't already, set up your application cluster to use the cache server address. You can find this in the Admin Console, when you go to **System** > **Settings** > **Caches**

Setting Up the Conversion Machine

The document conversion machine should be a dedicated machine that isn't used for any other Jive services. After you install Jive on the conversion machine, run `jive enable docconverter` as the **jive** user to disable all the services other than jive-docconverter.

Post-Installation Tasks

This section is intended to provide sample configurations and script examples common to long-term operation of a Jive installation.

Using Commands to Work with Your Managed Instance

Jive includes several command-line tools you can use to perform maintenance tasks with your managed instance. With these tools, you can start and stop the application, upgrade the application, collect information needed by Jive support, and more.

You'll find these documented in the Application Management Command Reference.

Enabling SSL Encryption

Enabling encryption of HTTP traffic requires several steps on a platform-managed host: you'll need to enable this for your required load balancer. For more information, see Configuring SSL on a Load Balancer.

Setting Up a Document Conversion Node

If you have purchased the Document Conversion module, see Setting Up a Document Conversion Node Some documents -- including PDFs and those from Microsoft Office -- are supported in a preview view in Jive. If you want to convert content from its native format into a form that can be previewed without altering the original document, you'll need the Document Conversion module, which you'll need to deploy on a server that is separate from your core Jive production instances.

Configuring the Text Extraction Location

For content search, all binary content uploaded to Jive, such as .doc, .ppt, .pdf, .txt, or .xls, goes through a process where Jive extracts the plain text from the documents so it can be used in the search index. By default, the output for this process is stored on the web app node in `/usr/local/jive/applications/sbs/home/search/search-text-extraction` (these are encrypted files). However, we strongly recommend you change this to an NFS-mounted directory on a different node. In clustered environments, the NFS directory must be shared by all web app nodes in the cluster.

To specify the new text extraction location, you need to set the `webapp.custom_jvm_args` by typing the following as root. Use `jive list set webapp.custom_jvm_args` to see if you have set this value. If you have you can add to the existing value.

```
jive set webapp.custom_jvm_args " -Djive.text.extract.dir=/path/to/new/
location"
```

Configuring Services Directory for On-Premise Search

When you install On-Premise search onto a multi-homed machine and you use the default host value of `0.0.0.0.`, On-Premise search may not choose the desired network interface. Therefore, if you are running On-Premise search on a multi-homed machine, you need to explicitly configure which network interface you want to bind to by changing the host values in the `serviceDirectory.json` file. For more on this, see Configuring Services Directory for On-Premise Search.

Installing On a Cluster

Before You Begin

Before you set up the nodes in a cluster, you should have already configured a cache server, as described in Setting Up a Cache Server. The cluster will require the presence of a cache server to cache data that should be available to all nodes in the cluster. If your cache server isn't configured and running, you won't be able to set up the cluster.

Note: Your license determines whether or not clustering is enabled and how many nodes are supported. To check on the number of clustered servers your license allows, see the license information after logging into the Admin Console.

Topology

The nodes in a cluster need to be installed on the same subnet, and preferably on the same switch. You cannot install nodes in a cluster across a WAN.

Upgrading

- **IMPORTANT** If you're upgrading and copying the home directory (such as `/usr/local/jive/applications/<instance_name>/home`) from the older installation, you must preserve the `node.id` file and the crypto directory from the home directory before starting the server. The value stored in this file must be unique among the cluster nodes; that is, each node in a cluster will have a unique value in the node.id file. You must preserve the `node.id` file because it plays a role in storing encrypted information in the cluster; if that file is lost, you will lose access to the encrypted information.

  If you are deploying a new cluster, it is permissible to copy the contents of the home directory from the first node (where you set up clustering) to subsequent nodes -- with the exception of the `node.id` file. Do not copy the `node.id` file to subsequent nodes. If the `node.id` file does not exist, the application will generate a new file on startup.
- The cache server must be cleared and restarted before the upgraded application server nodes are started and try to talk to the cache.
- If you're upgrading a plugin, clear the cache for that plugin and shut down the cache server first.

Starting a New Cluster

Always wait for the first node in the cluster to be up and running with clustering enabled before you start other cluster nodes. Waiting a minute or more between starting each node ensures the nodes are not in competition. As the senior member, the first node you start has a unique role in the cluster. See the clustering overview for more information.

Clocks

- The clocks on all machines must be synchronized for caching to work correctly. For more information, take a look at Managing Cache Servers. Also, if you're running in a virtualized environment, you must have VMware tools installed in order to counteract clock drift.
- If you're running in a virtualized environment, you must have VMware tools installed in order to counteract clock drift.

Cluster Node Communication

- Do not put a firewall between your cache servers and your Jive application servers. If you do so, caching will not work. A firewall is unnecessary because your application servers will not be sending

untrusted communications to the cache servers, or vice versa. There should be nothing that might slow down communication between the application servers and the cache servers.

- All ports between the cache and web application servers must be open.
- Port 6650 should be blocked to *external* access (but not between the cluster nodes!) so that any access outside of the datacenter is disallowed. This is to prevent operations allowed by JMX from being executed on the cache server.

Overview of a Cluster Installation

1. Be sure to read the System Requirements for important information about software, hardware, and network requirements and recommendations.
2. Provision a database server. Be sure to read the Database Prerequisites.
3. If you're going to use a separate server for binary storage, Configure a Binary Storage Provider.
4. If your community will use the document conversion feature, see Setting Up a Document Conversion.
5. Install a cache server on a separate server.
6. Install and configure the application on the first node in your cluster.
7. Install and configure the application on the subsequent nodes in your cluster.

Installing on a Cluster

Important: If, as part of your new installation, you're setting up one node as a template, then copying the home directory (such as `/usr/local/jive/applications/`*`your_instance_name`*`/home`) to other nodes in the cluster, you must **remove** the `node.id` file and the crypto directory from the home directory before starting the server. The application will correctly populate them.

1. Use the Jive application package to set up a cache server on a separate machine. See Setting Up a Cache Server for more information. Note the cache server address for use in setting up the application servers.
2. Before proceeding, make sure the cache server you set up is running. It must be running while you set up the application server nodes.
3. On each node in the cluster, install the application using the package (RPM on Linux), but don't run the Admin Console's setup wizard.

   See the Linux installation instructions for more information on installing the application.
4. Start the primary node and navigate to its instance with a web browser. In the setup screen provided, enter the address of the cache server you installed, then complete the Admin Console setup wizard.
5. After you've finished with the setup wizard, restart the node.
6. Copy the `jive.license` file, the `jive_startup.xml` file, and the `search` folder from the `home` directory on the primary node to the `home` directory in each of the other nodes in the cluster. The home directory is typically found here: `/usr/local/jive/applications/`*`your_instance_name`*`/home`.

7. On each of the secondary nodes, **remove** the `node.id` file and the crypto directory from the home directory. (The application will correctly populate these on each node when they are started for the first time.)
8. Start the application on each of the secondary nodes (`service jive-application start` followed by `service jive-httpd start`). Because they are connecting to the same database used by the primary server, each secondary node will detect that clustering is enabled and will pick up the configuration you set on the primary node.
9. Restart all the servers in the cluster to ensure that the address of each node in the cluster is known to all the other nodes.

Configuring a Cluster Node (optional)

You can get information about any node in the cluster and make configuration changes to it from the Admin Console. You can also enable or disable clustering from a node in the cluster and set the node's cluster address (which is unnecessary unless you want to ensure that the node has a particular TCP endpoint--IP address and port number). By design, a node that is new to a cluster creates its own address and registers itself in the database as a member of the cluster. However, you need to manually copy the encryption keys from the `/usr/local/jive/applications/`*`app_name`*`/home/crypto` directory to the new node before you can start it successfully.

Be sure to read the clustering overview for information about how the clustering system works.

Fastpath: Admin Console: System > Settings > Cluster

Use the following settings on the Admin Console's Cluster page to get information and configure the cluster:

Setting	Description
Enable cluster	Click **Enabled** to enable this node for the cluster.
Cluster members	Lists the addresses of other nodes in this cluster. Select the Remove check box to have this node's address removed from the database.
Local Cluster Address	Displays this node's IP address and the port on which this node listens for others in the cluster. The IP address and port form the unique TCP endpoint for this node in the cluster.
Cluster Overview	Lists the nodes in this cluster.

Adding Features with Plugins

By purchasing optional plugins offered by Jive Software, you can enhance Jive with powerful features not included by default. You can learn more about supported plugins at the Jive website.

Configuring Delegated Authentication

In delegated authentication, Jive delegates authentication to your user identity provider.

Note: For information on building delegated authentication support for your user identity provider, see Jive Delegated Authentication in the Jive Community.

Fastpath: Admin Console: People > Settings > Delegated Authentication Settings

Use the following high-level steps to understand the configuration process. The sections below provide more information on the settings themselves.

1. Select the **Enable Delegated Authentication** check box to reveal other configuration options.
2. Under **Services**, select the services for which you want authentication delegated.
3. Under **Options**, select optional features to go along with authentication.
4. Under **Service Location**, enter the URL at which your authentication web service can be found.
5. Test communication from the application to your web service.
    a. Enter a user name and password that will provide access to the web service.
    b. Enter the IP address for this community.
    c. Click **Perform Test**.

Services

This section lists services provided by the application, and which can require authentication for access. In other words, each of these represents a point of access for users. Select the services whose authentication requests should be delegated to the authentication provider you're describing in configuration here.

- Web interface -- The application's browser-based user interface. This is what your users will likely use most often.
- Mobile integration -- Access via a mobile device, such as the iPhone. Note that this option has been deprecated as of Jive v5.0. To enable delegated authentication for the Mobile plugin, select the **Web interface** option.
- Web services -- Access via SOAP- or REST-based web services.
- RSS feeds -- Access via RSS/Atom calls, such as from a feed aggregator.

Options

These are optional actions you can have the delegated authentication feature perform.

- Auto-create users -- Select this to have the application create internal user accounts for users it authenticates with your identity provider, but who aren't represented in the application's database yet.
- Synchronize profile fields -- Select this to synchronize user profiles between the application's profile data and profile data stored by your identity provider.

Service Location

The service address is the location at which to find your authentication web service.

- Username -- A user name known to the user identity provider.
- Password -- The password for the username provided.
- Source IP -- An optional field if your authentication web service will evaluate the IP address of the incoming request. For example, you might use this if you anticipate allowing access from only one IP address and you want to test that functionality here.

Configuring a Binary Storage Provider

By default, Jive stores binary content (attachments on blog posts, discussions, uploaded documents, images, profile pictures, and avatars) in the application database, but we recommend migrating binary content to a different binary storage provider. Storing binary content data in a location outside the database is more efficient and scalable. For production Jive systems, we recommend using a shared file system for storing all
binary content. Documents created in the content editor are
not binary content and are stored in the application database.

Preparing to Migrate to a Shared File System Location

You need to migrate your binary storage provider if:

- You want to use a file system provider instead of the JDBC provider enabled by default.
- You want to use the provider you're currently using, but change how you're using it. For example, you might want to use a different file system location for the file system provider.

Before you migrate to a different provider, be sure you do the following:

- Back up your current storage provider's backing store.
- Run a successful migration on a backup of this system.
- Do not run the migration tool during a busy period.
- Make sure the file system location is always available to the application.
- Estimate the amount of space you'll need for shared storage. For more on these requirements, see Binary Storage in Required External Components

Keep the following things in mind as you go through the migration process:

- During the migration, any new binary content will be stored in both the current and the new storage provider. Once the migration has completed successfully the system will switch over to the new provider.
- If any errors occur during migration the system will revert back to the current storage provider settings to allow the system to continue to function normally. Errors will be logged (see Getting Application Logs for more on logs). When you've addressed the errors, start over with migration.

- Keep in mind that binary content caching is disabled during migration. This might cause increased load on both the current storage provider as well as the application in general.
- Allow at least a couple of hours for binary storage migration unless you have a trivial amount of content, for example, if you're just starting a community.
- When configuring a file system provider, you'll specify a namespace. The namespace helps ensure that data is isolated for the application instance it belongs to. So, for example, if you had multiple instances (such as an internal and an external instance), you could use the same storage server while giving the storage provider for each instance a different namespace.

Migrating to a Shared File System for Binary Storage

Fastpath: Admin Console: System > Settings > Storage Provider

To migrate to a shared file system:

1. Prepare a single unit of shared NFS or block-level storage on your SAN/NAS, and mount it from each of your web application nodes using the same mount point on each node (for example, `/opt/jiveBinStore`). Refer to Sizing Binary Storage for help estimating the amount of shared storage you'll need.
2. From the Storage Provider page in the Admin Console, click **Migrate to another Storage Provider**.
3. Select **FileStorageProvider** in the **Registered Providers** field.
4. Click **Continue**.
5. Enter a namespace that will correspond to binary data for this application instance.
6. Enter the path to the local directory that you mapped to a mountable location in Step 1.
7. Click **Continue**.
8. Read the notes and confirm that the settings shown are the values you specified.
9. To delete binary content from the current source storage after migration, select the **Delete binary content** check box.
10. Click **Start Migration** to begin the process.
11. When the migration is complete, restart the application.
12. Disable local file caching:
    a. Navigate to **Admin Console**: **System** > **Settings** > **Storage Provider**
    b. Click **Edit** under Caching.
    c. Select **No** in the Cache Enable field.
    d. Click **Save**.

Jdbc Storage Providers

You can also migrate content to Jdbc storage providers (a DBMS).

To migrate to a different provider:

1. In the Admin Console go to the **Storage Provider** page.

2. Click **Migrate to another storage provider**.
3. In the Registered Providers field, select **JdbcStorageProvider**.

About Using a Database for Binary Storage

When you use another database for binary content storage, you specify a JNDI name corresponding to the database. During migration to the new provider, the application will create the database tables needed.

In a cluster, the JNDI name must be the same on each node where you configure an instance to use the provider.

Backup and Storage Considerations

Storage Reliability

You should mount the Jive system home directory (by default, this is `/usr/local/jive`) on redundant external storage (preferably SAN storage via redundant HBAs and SAN fabric). When redundant external storage is not available, the local system volume for your Jive installation directory should be mirrored across multiple physical disks to prevent the loss of a single disk as a single point of failure.

The total storage requirements for this directory will vary from installation to installation. As a basic guide for capacity planning, consider the following:

- Core binaries - The base installation requires 500MB storage (200MB on disk, an additional 300MB needed during upgrades of the platform).
- Total system traffic - The system writes all logs to *`your Jive installation directory`*/var/logs. While the system will by default rotate log files to reduce disk space consumed, larger installations may wish to retain log files for analysis over time (HTTPD access logs for example). In a default installation, allocating 5GB for log storage should provide ample room to grow.
- Cache efficiency - For each application, local caches of binary content including attachments and images are maintained. The more space available to those caches, the more efficient the system will be at serving binary requests and the smaller the strain on the backing RDBMS. As a capacity guideline, plan on roughly .25 the planned total binary (BLOB) storage in the RDBMS for efficient caching.
- Search index size - Each node stores local copies of the system search index. As a general rule of thumb, plan for search indexes to be 1x the total database storage consumption (.5 for active indexes, . 5 for index rebuilds).
- Local database backups - When using the Jive platform-managed database, the database will regularly be backed up to *`your Jive installation directory/`* `var/data/backup/full` and database checkpoint segments backed up to *`your Jive installation directory`*`/var/data/backup/wal`. When an instance is using this database, approximately 35x the total database size will be required in the *`your Jive installation directory`*`/var/data/backup` location with a default configuration. This number can be lowered by more aggressively removing full backup archives stored in backup/full and by more aggressively removing WAL segments after a full backup has been performed.

Storage Monitoring

As with any system, disk consumption should be regularly monitored and alerts generated when the system approaches disk capacity. Most disk consumption will occur in three areas:

- The application instance home directory -- By default, the platform manages a single application instance located at *your Jive installation directory*/applications/sbs with a home directory of sbs/home.
- Platform logs -- All platform log files are stored in *your Jive installation directory*/var/logs
- Platform database -- If the local platform database is used, data files will be stored by default in *your Jive installation directory* var/data/postgres-9.x and backups in *your Jive installation directory* /var/data/backup

System Backups

In addition to performing regular backups of reliable storage, you should perform the following backups:

1. All databases
2. Binary storage
3. The Jive installation directory (by default, this is /usr/local/jive).

The simplest solution is to back up the entire contents of the Jive installation directory. A more selective option is to back up only the /applications and /etc subdirectories. In either case, you should make backups in accordance with standard backup practices.

Before you upgrade Jive, you should make a full backup of the entire installation directory.

Quick Installation for a Jive Evaluation

Use this Quick Installation procedure to set up a small instance of Jive for evaluation and testing purposes only.

Note: The following steps assume you are installing Jive on a supported Linux operating system and you've never had Jive installed on the computer you're using for this evaluation. This procedure also assumes you are not running the Document Conversion feature or are tying any additional web application cluster or cache nodes to the environment.

Step	Task	More Details
1	Review the System Requirements.	Although this is an evaluation of Jive, you still need to install it on a supported Linux operating system and make sure it can communicate with remote resources. If you use a proxy server, make sure it's configured correctly to access the required external resources. To find out more, see Connecting to Jive-Hosted Services and Connecting Through a Proxy Server.

Step	Task	More Details
2	Copy the Jive application RPM package to all nodes in the environment.	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`
3	Install the Jive RPM package on all nodes.	Use a command such as **`rpm -Uhv jive.rpm`**. Note that your copy of the Jive RPM package (called jive.rpm in this example) will have a slightly different name.
4	Configure Jive using the setup wizard by running `jive setup` on all of the nodes.	Note that you must be the jive user to run `jive setup` (**`su - jive`**). You may need to re-run `jive setup` until you get the `setup ok` message.
5	Install and configure the Activity Engine Database, Core Databases, and the optional Analytics Database.	For evaluation purposes, you can install the Activity Engine, Analytics, and Core databases together, but this is not supported in a production environment. The simplest way to set up a database on Linux is to perform a Quick database setup for evaluation using a Postgres database. If you'll be using a database whose driver is not included, ensure its driver is in the application's classpath by following the steps in Database Prerequisites.
6	Start the services.	Note that you must be the jive user to run these commands (**`su - jive`**): • **`jive enable eae`** • **`jive enable search`** • **`jive enable webapp`** • **`jive enable httpd`**
7	Re-run `jive setup` on all of the nodes.	This causes the installer to detect any service-specific startup properties that need to be set. Run `jive setup` until you get the `setup ok` message.
8	After you see the `setup ok` message, start the instance.	**`jive start`**

Step	Task	More Details
9	From a web browser, visit your new community and finish configuring it from the Admin Console's setup wizard.	With a supported web browser, navigate to http://<hostname>:8080, where hostname is the DNS-resolvable name of the server where you installed the Jive application.
10	Integrate your Directory Server.	If you plan to populate your community with users synchronized from your LDAP or Active Directory implementation, see the LDAP and Active Directory Guide.