SIP Servlet-Enabled JBoss Application Server

Installing, Configuring and Running

Table of Contents

The Restcomm SIP Servlets Server can run on either the JBoss Application Server or the Tomcat Servlet Container. This section details how to install the SIP Servlets Server on top of the JBoss Application Server. For installation instructions for the Tomcat Servlet Container, refer to [_bssswticar_sip_servlets_server_with_tomcat_installing_configuring_and_running]

It is recommended that the SIP Servlets Server is run on the JBoss platform. Some functionality, including the ability to execute some SIP Extension examples, is not available in the Tomcat version.

Differences Between a Standard JBoss Installation and the Restcomm SIP Servlets Version

Provided here is a list of differences between a standard JBoss Application Server installation one customized for SIP Servlets. The differences include:

- The *server/default/deploy* directory contains both HTTP and SIP Servlet applications.
- The *server/default/deploy/jbossweb.sar* units have been modified to provide extended classes to the standard JBoss container classes, in order to allow SIP applications to be loaded and the SIP stack to be started.
- The *server/default/deploy/jbossweb.sar context.xml* files have been modified to allow the extended manager to manage SIP sessions and SIP application sessions in addition to HTTP sessions.
- The *server/default/deploy/jbossweb.sar/ server.xml* file has been modified to provide extended classes to common JBoss Web containers. The classes allow SIP applications to be loaded, and the SIP stack to be started.
- The *server/default/deploy/jbossweb.sar/ jboss-beans.xml* file has been modified to allow the JBoss container to process SIP messages.
- The *server/default/deployers/ metadata-deployer-jboss-beans.xml* file has been modified to allow JBoss to parse sip.xml deployment descriptors and SIP metadata annotations.
- The *server/default/deployers/jbossweb.deployer/META-INF/war-deployers-jboss-beans.xml* file have been modified so that it is now possible for JBoss containers to correctly deploy SIP servlets and converged applications.
- A *dars* directory containing all of the Default Application Router (DAR)

properties files for using the various SIP Servlets applications (which come bundled with the release) has been added to the *server/default/conf* directory.

- Additional JAR files have been added to enable SIP Servlet functionality; these are located in the *server/default/deploy/jbossweb.sar/*, *server/default/deployers/jbossweb.deployer/* and [path]_common/lib_directories.

Java Development Kit (): Installing, Configuring and Running

The [app]` Platform` is written in Java; therefore, before running any `server, you must have a working Java Runtime Environment () or Java Development Kit () installed on your system. In addition, the JRE or JDK you are using to run [app]` must be version 5 or higher [1: At this point in time, it is possible to run most servers, such as the JAIN SLEE Server, using a Java 6 JRE or JDK. Be aware, however, that presently the XML Document Management Server does not run on Java 6. We suggest checking the web site, forums or discussion pages if you need to inquire about the status of running the XML Document Management Server with Java 6.].

Should I Install the JRE or JDK?

Although you can run `servers using the Java Runtime Environment, we assume that most users are developers interested in developing Java-based, [app]`-driven solutions. Therefore, in this guide we take the tact of showing how to install the full Java Development Kit.

Should I Install the 32-Bit or the 64-Bit JDK, and Does It Matter?

Briefly stated: if you are running on a 64-Bit Linux or Windows platform, you should consider installing and running the 64-bit JDK over the 32-bit one. Here are some heuristics for determining whether you would rather run the 64-bit Java Virtual Machine (JVM) over its 32-bit cousin for your application:

- Wider datapath: the pipe between RAM and CPU is doubled, which improves the performance of memory-bound applications when using a 64-bit JVM.
- 64-bit memory addressing gives virtually unlimited (1 exabyte) heap allocation. However large heaps affect garbage collection.
- Applications that run with more than 1.5 GB of RAM (including free space for garbage collection optimization) should utilize the 64-bit JVM.
- Applications that run on a 32-bit JVM and do not require more than minimal heap sizes will gain nothing from a 64-bit JVM. Barring memory issues, 64-bit hardware with the same relative clock speed and architecture is not likely to run Java applications faster than their 32-bit cousin.

Note that the following instructions detail how to download and install the 32-bit JDK, although the steps are nearly identical for installing the 64-bit version.

Downloading

You can download the Sun JDK 5.0 (Java 2 Development Kit) from Sun's website:

http://java.sun.com/javase/downloads/index_jdk5.jsp. Click on the Download link next to "JDK 5.0 Update <x>`" (where [replaceable]<x>` is the latest minor version release number). On the next page, select your language and platform (both architecture—whether 32- or 64-bit—and operating system), read and agree to the Java Development Kit 5.0 License Agreement, and proceed to the download page.

The Sun website will present two download alternatives to you: one is an RPM inside a self-extracting file (for example, *jdk-1_5_0_16-linux-i586-rpm.bin*), and the other is merely a self-extracting file (e.g. *jdk-1_5_0_16-linux-i586.bin*). If you are installing the JDK on Red Hat Enterprise Linux, Fedora, or another RPM-based Linux system, we suggest that you download the self-extracting file containing the RPM package, which will set up and use the SysV service scripts in addition to installing the JDK. We also suggest installing the self-extracting RPM file if you will be running [app]`` in a production environment.

Installing

The following procedures detail how to install the Java Development Kit on both Linux and Windows.

Procedure: Installing the JDK on Linux

1. Regardless of which file you downloaded, you can install it on Linux by simply making sure the file is executable and then running it:

```
~]$ chmod +x "jdk-1_5_0_<minor_version>-linux-<architecture>-rpm.bin"
~]$ ./"jdk-1_5_0_<minor_version>-linux-<architecture>-rpm.bin"
```

You Installed Using the Non-RPM Installer, but Want the SysV Service Scripts

If you download the non-RPM self-extracting file (and installed it), and you are running on an RPM-based system, you can still set up the SysV service scripts by downloading and installing one of the `-compat` packages from the JPackage project. Remember to download the `-compat` package which corresponds correctly to the minor release number of the JDK you installed. The compat packages are available from link:ftp://jpackage.hmdc.harvard.edu/JPackage/1.7/generic/RPMS.non-free/.

You do not need to install a `-compat` package in addition to the JDK if you installed the self-extracting RPM file! The `-compat` package merely performs the same SysV service script set up that the RPM version of the JDK installer does.

Procedure: Installing the JDK on Windows

1. Using Explorer, simply double-click the downloaded self-extracting installer and follow the instructions to install the JDK.

Configuring

Configuring your system for the JDK consists in two tasks: setting the `JAVA_HOME` environment variable, and ensuring that the system is using the proper JDK (or JRE) using the `alternatives` command. Setting `JAVA_HOME` usually overrides the values for `java`, `javac` and `java_sdk_1.5.0` in

`alternatives`, but we will set them all just to be safe and consistent.

Setting the `JAVA_HOME` Environment Variable on Generic Linux

After installing the JDK, you must ensure that the `JAVA_HOME` environment variable exists and points to the location of your JDK installation.

Setting `java`, `javac` and `java_sdk_1.5.0` Using the `alternatives` command

As the root user, call `/usr/sbin/alternatives` with the `--config java` option to select between JDKs and JREs installed on your system:

Setting the `JAVA_HOME` Environment Variable on Windows

For information on how to set environment variables in Windows, refer to http://support.microsoft.com/kb/931715.

Testing

Finally, to make sure that you are using the correct JDK or Java version (5 or higher), and that the java executable is in your `PATH`, run the `java -version ` command in the terminal from your home directory:

```
~]$ java -version
java version "1.5.0_16"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_16-b03)
Java HotSpot(TM) Client VM (build 1.5.0_16-b03, mixed mode, sharing)
```

Uninstalling

There is usually no reason (other than space concerns) to remove a particular JDK from your system, given that you can switch between JDKs and JREs easily using `alternatives`, and/or by setting `JAVA_HOME`.

Uninstalling the JDK on Linux

On RPM-based systems, you can uninstall the JDK using the `yum remove <jdk_rpm_name> ` command.

Uninstalling the JDK on Windows

On Windows systems, check the JDK entry in the `Start` menu for an uninstall command, or use `Add/Remove Programs`. :leveloffset: 0

Pre-install Requirements and Prerequisites

Hardware Requirements

Sufficient Disk Space

Once unzipped, version &VERSION;of the Restcomm SIP Servlets for JBoss binary release requires a minimum of &SSS_MSS4J_SIZE;free disk space.

Anything Java Itself Will Run On

Restcomm SIP Servlets for JBoss is 100% Java and will run on the same hardware that the JBoss Application Server runs on.

Software Prerequisites

JDK 5 or Higher

A working installation of the Java Development Kit (<acronym>JDK</acronym>) version 5 or higher is currently required in order to run Restcomm SIP Servlets for JBoss binary distribution. For instructions on how to install the JDK, refer to Installing

Downloading

The latest version of Restcomm SIP Servlets for JBoss is available from &DOWNLOAD_LINK;link:.

Each version of the SIP Servlets Server is comprised of tree separate binary distribution files: one which is Restcomm SIP Servlets for JBoss, one which is Restcomm SIP Servlets for EAP and the other which is Restcomm SIP Servlets for Tomcat. Download SIP Servlets Server for JBoss and continue with the following instructions.

Installing

Once the requirements and prerequisites have been met and you have downloaded the binary distribution zip file, you are ready to install the Restcomm SIP Servlets for JBoss binary distribution. Follow the instructions below for the selected platform, whether Linux or Windows.

Version Numbers

For clarity, the command line instructions presented in this chapter use specific version numbers and directory names. Ensure this information is substituted with the binary distribution's version numbers and file names.

Procedure: Installing the Restcomm SIP Servlets for JBoss Binary Distribution on Linux

It is assumed that the downloaded archive is saved in the home directory, and that a terminal window is open displaying the home directory . Create a subdirectory to extract the Restcomm SIP Servlets for JBoss files into. For ease of identification, it is recommended that the version number of the binary is included in this directory name.

+

```
~]$ mkdir "-jboss-<version>"
```

1. Move the downloaded zip file into the directory.

   ```
   ~]$ mv "" "-jboss-<version>"
   ```

2. Move into the directory.

```
~]$ cd "-jboss-<version>"
```

3. Extract the files into the current directory by executing one of the following commands.

   Java:

```
-jboss-<version>]$ jar -xvf ""
* ----
Linux:
```

-jboss-<version>]$ unzip "" * ----

+ NOTE: You can also use `unzip'-d <unzip_to_location> to extract the zip file's contents to a location other than the current directory.

1. To free disk space, you may want to delete the zip file once you've extracted its contents:

```
-jboss-<version>]$ rm ""
```

Procedure: Installing the Restcomm for JBoss Binary Distribution on

For this procedure, it is assumed that the downloaded archive is saved in the *My Downloads* folder. . Create a directory in *My Downloads* to extract the zip file's contents into. For ease of identification, it is recommended that the version number of the binary is included in the folder name. For example, *-jboss-<version>*. . Extract the contents of the archive, specifying the destination folder as the one created in the previous step. . Alternatively, execute the `jar -xvf` command to extract the binary distribution files from the zip archive.

+ . Move the downloaded zip file from *My Downloads* to the folder created in the previous step. . Open the Windows Command Prompt and navigate to the folder that contains the archive using the `cd` command . Execute the `jar -xvf` command to extract the archive contents into the current folder.

+

```
C:\Users\<user>\My Downloads\-jboss-<version>>jar -xvf ""
```

1. It is recommended that the folder holding the Restcomm for JBoss files (in this example, the folder named *-jboss-<version>*) is moved to a user-defined location for storing executable programs. For example, the *Program Files* folder.
2. Consider deleting the archive, if free disk space is an issue.

```
C:\Users\<user>\My Downloads\-jboss-<version>>delete ""
```

Setting the JBOSS_HOME Environment Variable

Configuring Restcomm for JBoss consists of setting the JBOSS_HOME environment variable and optionally customizing the Restcomm for JBoss server by adding SIP Connectors, configuring the application router, and logging.

After setting JBOSS_HOME according to the instructions in the following section, refer to [_bsssc_binary_sip_servlets_server_configuring] to learn how to configure Restcomm for JBoss.

Alternatively, after having set JBOSS_HOME, the Restcomm for JBoss server can be run. Return to this section to configure it later.

Setting the JBOSS_HOME Environment Variable

The [app]` Platform` (`) is built on top of the [app]`JBoss Application Server (JBoss AS). You do not need to set the JBOSS_HOME environment variable to run any of the [app]` Platform` servers *unless* JBOSS_HOME is *already* set.

The best way to know for sure whether JBOSS_HOME was set previously or not is to perform a simple check which may save you time and frustration.

Checking to See If JBOSS_HOME is Set on Unix

At the command line, echo $JBOSS_HOME to see if it is currently defined in your environment:

```
~]$ echo $JBOSS_HOME
```

The [app]` Platform` and most Restcomm SIP Servlets servers are built on top of the JBoss Application Server (JBoss AS). When the [app]` Platform` or Restcomm SIP Servlets servers are built *from source*, then JBOSS_HOME *must* be set, because the Restcomm SIP Servlets files are installed into (or "over top of" if you prefer) a clean JBoss AS installation, and the build process assumes that the location pointed to by the JBOSS_HOME environment variable at the time of building is the JBoss AS installation into which you want it to install the Restcomm SIP Servlets files.

This guide does not detail building the [app]` Platform` or any Restcomm SIP Servlets servers from source. It is nevertheless useful to understand the role played by JBoss AS and JBOSS_HOME in the Restcomm SIP Servlets ecosystem.

The immediately-following section considers whether you need to set JBOSS_HOME at all and, if so, when. The subsequent sections detail how to set JBOSS_HOME on Unix and Windows

Even if you fall into the category below of *not needing* to set JBOSS_HOME, you may want to for various reasons anyway. Also, even if you are instructed that you do *not need* to set JBOSS_HOME, it is good practice nonetheless to check and make sure that JBOSS_HOME actually *isn't* set or defined on your system for some reason. This can save you both time and frustration.

You *DO NOT NEED* to set JBOSS_HOME if…

- ...you have installed the [app]` Platform` binary distribution.
- ...you have installed a Restcomm SIP Servlets server binary distribution *which bundles* JBoss AS.

You *MUST* set JBOSS_HOME if...

- ...you are installing the [app]` Platform` or any of the Restcomm SIP Servlets servers *from source*.
- ...you are installing the [app]` Platform` binary distribution, or one of the Restcomm SIP Servlets server binary distributions, which *do not* bundle JBoss AS.

Naturally, if you installed the [app]` Platform` or one of the Restcomm SIP Servlets server binary releases which *do not* bundle JBoss AS, yet requires it to run, then you should install JBoss AS before setting JBOSS_HOME or proceeding with anything else.

Setting the JBOSS_HOME Environment Variable on Unix

The JBOSS_HOME environment variable must point to the directory which contains all of the files for the [app]` Platform` or individual Restcomm SIP Servlets server that you installed. As another hint, this topmost directory contains a *bin* subdirectory.

Setting JBOSS_HOME in your personal *~/.bashrc* startup script carries the advantage of retaining effect over reboots. Each time you log in, the environment variable is sure to be set for you, as a user. On Unix, it is possible to set JBOSS_HOME as a system-wide environment variable, by defining it in */etc/bashrc*, but this method is neither recommended nor detailed in these instructions.

Procedure: To Set JBOSS_HOME on Unix...

1. Open the *~/.bashrc* startup script, which is a hidden file in your home directory, in a text editor, and insert the following line on its own line while substituting for the actual install location on your system:

```
export JBOSS_HOME="/home/<username>/<path>/<to>/<install_directory>"
```

2. Save and close the *.bashrc* startup script.
3. You should source the *.bashrc* script to force your change to take effect, so that JBOSS_HOME becomes set for the current session [2: Note that any other terminals which were opened prior to your having altered .bashrc will need to source ~/.bashrc as well should they require access to JBOSS_HOME.].

```
~]$ source ~/.bashrc
```

4. Finally, ensure that JBOSS_HOME is set in the current session, and actually points to the correct location:

The command line usage below is based upon a binary installation of the [app]` Platform`. In this sample output, `JBOSS_HOME` has been set correctly to the `topmost_directory` of the `installation. Note that if you are installing one of the standalone [app]` servers (with `JBoss AS` bundled!), then `JBOSS_HOME` would point to the `topmost_directory` of your server installation.

```
~]$ echo $JBOSS_HOME
/home/silas/
```

Setting the JBOSS_HOME Environment Variable on Windows

The `JBOSS_HOME` environment variable must point to the directory which contains all of the files for the Restcomm SIP Servlets Platform or individual Restcomm SIP Servlets server that you installed. As another hint, this topmost directory contains a *bin* subdirectory.

For information on how to set environment variables in recent versions of Windows, refer to http://support.microsoft.com/kb/931715. :leveloffset: 0

Configuring

To configure Restcomm for JBoss, refer to [_bsssc_binary_sip_servlets_server_configuring].

Running

To start the server, execute one of the startup scripts in the *bin* directory (on Linux or Windows), or by double-clicking the *run.bat* executable batch file in that same directory (on Windows only). It is recommended that the JBoss Application Server is started using the terminal or Command Prompt because the messages displayed during startup can be used to debug, and subsequently correct, any problems. In the Linux terminal or Command Prompt, a successfully started server will return the following information :

```
22:39:07,598 INFO  [SipApplicationDispatcherImpl]  Sip Servlets <version> - revision
<rev number> started.
```

Detailed instructions are given below, arranged by platform.

Procedure: Running Restcomm for JBoss on Linux

1. Change the working directory to Restcomm for JBoss's installation directory (the one in which the zip file's contents was extracted to)

```
downloads]$ cd "-jboss-<version>"
```

2. (Optional) Ensure that the *bin/run.sh* start script is executable.

```
-jboss-<version>]$ chmod +x bin/run.sh
```

3. Execute the *run.sh* Bourne shell script.

```
-jboss-<version>]$ ./bin/run.sh
```

Instead of executing the Bourne shell script to start the server, the *run.jar* executable Java archive can be executed from the *bin* directory:

```
-jboss-<version>]$ java -jar bin/run.jar
```

Procedure: Running Restcomm for JBoss on

There are several ways to start Restcomm for JBoss on Windows. All of the following methods accomplish the same task. . Using Windows Explorer, navigate to the *bin* subdirectory in the installation directory. . The preferred way to start Restcomm for JBoss from the Command Prompt. The command line interface displays details of the startup process, including any problems encountered during the startup process.

+ Open the Command Prompt via the Start menu and navigate to the correct folder:

+

```
C:\Users\<user>My Downloads> cd "-jboss-<version>"
```

Start the JBoss Application Server by executing one of the following files:

- *run.bat* batch file:

```
C:\Users\<user>My Downloads\-jboss-<version>>bin\run.bat
```

- *run.jar* executable Java archive:

```
C:\Users\<user>My Downloads\-jboss-<version>>java -jar bin\run.jar
```

Using

Once the server is running, access the SIP Servlets Management Console by opening http://localhost:8080/sip-servlets-management/.

Testing

After installation, there should be one pre-configured sample application deployed in the `default` server onfiguration. You can use it to verify that the server is installed and running correctly. The application name is “org.mobicents.servlet.sip.example.SimpleApplication”. From the Sip Servlets Management Console you can make sure it is subscribed to receive `INVITE` and `REGISTER` SIP requests. It is a simple *Click2Call* application allowing SIP registration and calling phones from the Web user interface.

The scenario for this example consists of the following steps:

1. Alice and Bob each register a SIP Softphone
2. Alice clicks on the "Call" link to place a call to Bob
3. Alice’s phone rings
4. When Alice picks up her phone, Bob’s phone rings
5. When Bob answers his phone, the call is connected
6. When one of them hangs up, the other one is also disconnected

Procedure: Testing the Click2Call sample application

1. Open up a browser to http://localhost:8080/click2call/. If you have no registered SIP clients you will be asked to register at least two.
2. Configure your SIP clients to use the sip servlets server as a register and proxy. (IP address : 127.0.0.1, port: 5080) By default it will accept any password
3. After the registration you will see a table where each cell will initiate a call between the corresponding clients.
4. Close the calls.
5. Navigate to http://localhost:8080/click2call/simplecall.html, which is a simplified version that doesn’t require registered clients.
6. Enter the URIs of the two SIP phones you just started and click "Submit"
7. The phones should be ringing again. You can pick them up and you will know that the SIP and the HTTP containers are working properly.

Stopping

Detailed instructions for stopping the JBoss Application Server are given below, arranged by platform. If the server is correctly stopped, the following three lines are displayed as the last output in the Linux terminal or Command Prompt:

```
[Server] Shutdown complete
Shutdown complete
Halting VM
```

Procedure: Stopping Restcomm for JBoss on Linux

1. Change the working directory to the binary distribution's install directory.

```
~]$ cd "-jboss-<version>"
```

2. (Optional) Ensure that the bin/shutdown.sh start script is executable:

```
-jboss-<version>]$ chmod +x bin/shutdown.sh
```

3. Run the *shutdown.sh* executable Bourne shell script with the `-S` option (the short option for `--shutdown`) as a command line argument:

```
-jboss-<version>]$ ./bin/shutdown.sh -S
```

The *shutdown.jar* executable Java archive with the `-S` option can also be used to shut down the server:

```
-jboss-<version>]$ java -jar bin/shutdown.jar -S
```

Procedure: Stopping &CMD_PLATFORM_NAME; for JBoss on Windows

1. Stopping the JBoss Application Server on Windows consists in executing either the *shutdown.bat* or the *shutdown.jar* executable file in the *bin* subdirectory of the Restcomm for JBoss binary distribution. Ensure the `-S` option (the short option for `--shutdown`) is included in the command line argument.

```
C:\Users\<user>\My Downloads\-jboss-<version>>bin\shutdown.bat -S
```

   a. The *shutdown.jar* executable Java archive with the `-S` option can also be used to shut down the server:

```
C:\Users\<user>\My Downloads\-jboss-<version>>java -jar bin\shutdown.jar -S
```

Uninstalling

To uninstall Restcomm for JBoss, delete the directory containing the binary distribution.