



THE
POWER
TO KNOW.

Third-Party
Software
Support

Updating an Existing SAS[®] 9.3 Enterprise BI Server (TS1M2) Deployment to Use JBoss Application Server 7.1

Table of Contents

Overview	1
Step 1: Download and Extract JBoss AS 7.1.1	1
Step 2: Update the JBoss and JDK Entries in the SAS Deployment Registry	2
Step 3: Execute Scripts That Modify Your Existing Configuration to Reference JBoss AS 7.1.1	2
Step 4: Make Required Modifications to Specific SAS Web Applications	3
Modify the JNDI Values in the sas.wip.services.war File	4
Modify JAR Files.....	4
Extract the JAR Contents.....	4
Modify the sasportal.tld File	5
Re-Create the JAR File	5
Modify the portlet-container-was-taglib.tld File.....	5
Modify the JNDI Values in the sas.workflow.war File.....	6
Copy and Rename the jboss-deployment-structure.xml_forworkflow and the jboss-deployment-structure.xml_forwipservices Files	6
Modify the JNDI Values in the sas.wip.soapservices.war File	6
Remove the Unregistered Security Domain in JBoss AS 7.1.1	7
Modify the sas.portlet.container.client.jar File	7
Modify Cookie Settings.....	8
Step 5: Start JBoss AS 7.1.1. and Validate SAS Web Applications	9

Overview

This document explains how to upgrade your existing SAS® 9.3 Enterprise BI Server installation that uses the JBoss Application Server (AS) 4.2 or 5.1 so that it uses JBoss AS 7.1.

Note: For the minimum JBoss version required for use with SAS 9.3, see “JBoss Application Server Support for SAS® 9.3” (support.sas.com/resources/thirdpartysupport/v93/appservers/jbossw.html)

This process is for SAS Enterprise BI Server at the second maintenance release for SAS® 9.3 (TS1M2). The document provides prerequisite installation and configuration steps (including modifications you need to make to properties file), steps for running the upgrade script, and SAS web application modifications that help you to avoid problems that can occur with SAS Enterprise BI Server web applications that are deployed to JBoss AS 7.1.

Notes:

- **This document addresses only modifications that are required to run SAS Enterprise BI web applications.** If your organization deploys any solutions besides SAS Enterprise BI Server, contact your local SAS Technical Support office for information about upgrading to JBoss AS 7.1.
- SAS 9.3 TS1M1 or earlier only supports JBoss AS 4.2.3 and JBoss AS 5.1.0 using Java Development Kit (JDK) 6. SAS 9.3 TS1M2 runs with JBoss AS 7.1.1. The system requirements for JBoss AS 7.1.1 include both JDK 6 and JDK 7. However, SAS Technical Support **strongly recommends** that you do not upgrade to JBoss AS 7.1.1 unless you have a specific business need that requires you to upgrade to Java 7.
- Steps in this document do not address upgrading to Java 7. For more details, see SAS Note 49296, “Using Java 7 with SAS® 9.3 TS1M2.” (support.sas.com/kb/49/296.html)

Step 1: Download and Extract JBoss AS 7.1.1

1. Download the jboss-as-7.1.1.Final.zip file from the “JBoss Application Server 7 Downloads” page (www.jboss.org/jbossas/downloads/) by clicking the ZIP download link for the 7.1.1.Final version of the application server.
2. Unzip jboss-as-7.1.1.Final.zip into a new installation location of your choice.

Step 2: Update the JBoss and JDK Entries in the SAS Deployment Registry

1. On your middle-tier host, back up the registry.xml file that resides in *SAS-installation-directory/deploymenreg/*.
2. Open the registry.xml file in your text editor and modify the JBoss and JDK locations shown below in red so that they match your environment. The following code shows an example:

```
<Key name="jboss">
  <Key name="default">
    <Value data="09D2PR" name="order"/>
    <Value data="JBoss Application Server" name="displayname"/>
    <Value data="jboss-installation-directory" name="location"/>
    /* In the line above, replace jboss-installation-directory */
    /* with the path to your specific JBoss directory (for */
    /* example, C:/jboss-as-7.1.1.Final). */
  </Key>
</Key>

<Key name="jdk">
  <Key name="default">
    <Value data="09D2PR" name="order"/>
    <Value data="Java Development Kit" name="displayname"/>
    <Value data="jdk-home-directory" name="location"/>
    /* In the line above, replace jdk-home-directory with your */
    /* specific JDK directory path (for example, C:/jdk1.7.0_15) */
  </Key>
</Key>
```

Step 3: Execute Scripts That Modify Your Existing Configuration to Reference JBoss AS 7.1.1

1. First, you need to download the **JBoss7Upgrade** ZIP file to a directory of your choice on your machine. To download the ZIP file, see SAS Note [49775](#), "Required manual steps for upgrading to JBoss Application Server (AS) 7.1 with SAS® 9.3 (TS1M2)."
2. From your **JBoss7Upgrade** directory, copy the **7.1.1** directory to *SAS-configuration-directory\Levn\Web\Scripts\JBoss*.
3. Copy the jbossScripting47.bat and jbossScripting47.groovy script files that reside in **JBoss7Upgrade** to *SAS-configuration-directory\Levn\Web\Scripts\JBoss*.
4. Modify jbossScripting47.bat by changing the paths for the JBOSS_HOME=, JAVA_HOME=, and SCRIPT_DIR= parameters so that they reference your specific directory locations.

Example:

```
set JBOSS_HOME=C:\jboss-as-7.1.1.Final
set JBOSS_VERSION=7.1.1
set JAVA_HOME=C:\jdk1.7.0_15
set SCRIPT_DIR=SAS-configuration-directory\Levn\Web\Scripts\JBoss
```

Note: *Levn* specifies your configuration level.

5. Back up your `jbossScripting.properties` file, which is located in ***SAS-configuration-directory\Levn\Web\Scripts\JBoss***.

6. Edit `jbossScripting.properties`, as follows:

- Change the paths for `CONFIG.JBOSS.INSTALL.DIR=` and `CONFIG.JDK.INSTALL.DIR=` to reference your specific directory locations, as shown in the following example:

```
config.jboss.install.dir=C:\\jboss-as-7.1.1.Final
config.jdk.install.dir=C:\\jdk1.7.0_15
```

- Change the value for `CONFIG.APPSERVER.VERSION=` as shown below:

```
config.appserver.version=7.1.1
```

- Change the Java Virtual Machine (JVM) options (`-Djboss.home.dir=`, `-Djboss.server.home.dir=`, `-Dsas.deploy.dir=`, and `-Djava.security.policy=`) for each managed server in your deployment. The following example pertains to a deployment that has two managed servers. Your deployment might have more or fewer managed servers than the example.

```
server.server1.options=. . .-Djava.security.policy\\"C:\\jboss-as-7.1.1.Final\\server\\SASServer1\\conf\\sas.all.permissions.policy";-
Djboss.home.dir\\"C:\\jboss-as-7.1.1.Final";-
Djboss.server.home.dir\\"C:\\jboss-as-7.1.1.Final\\server\\SASServer1";
-Dsas.deploy.dir\\"C:\\jboss-as-7.1.1.Final\\standalone\\
deployments\\deploy_sas\\SASServer1"
. . .more option lines. . .
```

```
server.server2.options=. . .-Djava.security.policy\\"C:\\jboss-as-7.1.1.Final\\server\\SASServer2\\conf\\sas.all.permissions.policy";-
Djboss.home.dir\\"C:\\jboss-as-7.1.1.Final";-
Djboss.server.home.dir\\"C:\\jboss-as-7.1.1.Final\\server\\SASServer2";-Dsas.deploy.dir\\"C:\\jboss-as-7.1.1.Final\\standalone\\deployments\\deploy_sas\\SASServer2"
. . .more option lines. . .
```

7. Copy all of the files from the ***groovy*** directory (which is located in your ***JBoss7upgrade*** directory) to ***SAS-configuration-directory\Levn\Web\Scripts\JBoss\src\com\sas\config\jboss***.

8. From your ***SAS-configuration-directory\Levn\Web\Scripts\JBoss*** directory, run the `jbossScripting47.bat` script. If a password prompt appears for `sastrust@saspw` or `sasadm@saspw`, enter the correct password.

Step 4: Make Required Modifications to Specific SAS Web Applications

Before you can continue with the final validation step for your upgrade, you need to manually update certain SAS web applications. This section provides the steps for those updates.

Modify the JNDI Values in the sas.wip.services.war File

The sas.wip.services.war file is an important web application in SAS Enterprise BI Server because many other web applications are dependent on that file. However, the current Java Naming and Directory Interface (JNDI) values in sas.wip.services.war do not match the JNDI values that are defined in JBoss AS 7.1.1. Modify the values in sas.wip.services.war so that they match the values that are defined in JBoss AS 7.1.1, as follows:

1. Locate the jboss-web.xml file that resides in the *JBoss-home-directory*\standalone\deployments\deploy_sas\SASServer1\sas.wip.services9.3.ear\sas.wip.services.war\WEB-INF\.
2. In the jboss-web.xml file, add the prefix `java: /` to all of the Java Message Service (JMS) `<jndi-name>` attributes, as shown in this example:

```
<jndi-name>java:/sas/jms/AlertQueue</jndi-name>
```

3. Then add the prefix `java:jboss/` to the `DataSource` JNDI, as shown in this example:

```
<jndi-name>java:jboss/sas/jdbc/SharedServices</jndi-name>
```

Modify JAR Files

Modify the file named sasportal.tld in each of the following JAR files:

- *JBoss-home-directory*\standalone\deployments\deploy_sas\SASServer1\sas.portal4.3.ear\sas.portal.war\WEB-INF\lib\sas.portal.private.jar
- *JBoss-home-directory*\standalone\deployments\deploy_sas\SASServer1\sas.bidashboard4.3.ear\sas.eventsgenerationframework.war\WEB-INF\lib\sas.portal.jar
- *JBoss-home-directory*\standalone\deployments\deploy_sas\SASServer1\sas.bidashboard4.3.ear\sas.bidashboard.war\WEB-INF\lib\sas.portal.jar
- *JBoss-home-directory*\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.jsr168remoteportlet4.3.war\WEB-INF\lib\sas.portal.private.jar
- *JBoss-home-directory*\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.biportlets4.3.war\WEB-INF\lib\sas.portal.private.jar

For each JAR file listed, perform the steps in the following sections.

Extract the JAR Contents

1. Create a directory named `update.jar` in any directory outside of *JBoss-home-directory*.
2. Copy the JAR file that you want to modify to the `update.jar` directory.
3. From a command window, navigate to the `update.jar` directory.

- Submit the following command to extract the JAR file:

```
prompt: C:\jdk1.7.0_15\bin\jar xvf jar-name
```

Example: c:\java\jdk1.7.0_15\bin\jar xvf sas.portal.jar

- Delete the original JAR file from their respective directories. **Do NOT delete** the JAR files from the `update.jar` directory.

Modify the sasportal.tld File

- Open the `sasportal.tld` file, which resides in the `update.jar\META-INF>taglib` directory.
- In `sasportal.tld`, change each instance of `<bodycontent>jsp</bodycontent>` to `<bodycontent>JSP</bodycontent>` and change each instance of `<body-content>jsp</body-content>` to `<body-content>JSP</body-content>`. Then save `sasportal.tld`.

Note: There might be only one occurrence of `<body-content>jsp</body-content>` to modify in the `sasportal.tld` file for each JAR file.

Re-Create the JAR File

- Submit the following command to combine the extracted files and directories:

```
prompt: C:\jdk1.7.0_15\bin\jar cvfm jar-name META-INF\MANIFEST.MF *
```

Example: C:\jdk1.7.0_15\bin\jar cvfm sas.portal.private.jar
META-INF\MANIFEST.MF *

- Copy the newly created JAR file from `update.jar` to the `lib` directory of your web application.

Sample Path for lib Directory: C:\jboss-as-7.1.1.Final\standalone\deployments\
deploy_sas\SASServer1\sas.portal4.3.ear\
sas.portal.war\WEB-INF\lib

Modify the portlet-container-was-taglib.tld File

Change the values for the `<bodycontent>` parameter in `portlet-container-was-taglib.tld` so that the `sas.biportlets4.3.war` file can start properly. Change the content, as follows:

- Locate the `portlet-container-was-taglib.tld` file that resides in `JBoss-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.biportlets4.3.war\WEB-INF>taglib`.
- In that file, change each instance of `<bodycontent>jsp</bodycontent>` to `<bodycontent>JSP</bodycontent>`.

Modify the JNDI Values in the sas.workflow.war File

The JNDI values in sas.workflow.war do not match the JNDI values that are defined in JBoss AS 7.1.1. Modify these values in sas.workflow.war directory so that they match the values that are defined in JBoss, as follows:

1. Locate the jboss-web.xml file that resides in *JBoss-directory-home\standalone\deployments\deploy_sas\SASServer1\sas.workflow9.3.ear\sas.workflow.war\WEB-INF*.
2. In the jboss-web.xml file, add the prefix `java: /` to all of the JMS resource `<jndi-name>` attributes, as shown in this example:

```
<jndi-name>java:/sas/jms/QueueConnectionFactory</jndi-name>
```

3. Add the prefix `java:jboss/` to the DataSource JNDI, as shown in this example:

```
<jndi-name>java:jboss/sas/jdbc/SharedServices</jndi-name>
```

Copy and Rename the jboss-deployment-structure.xml_forworkflow and the jboss-deployment-structure.xml_forwipservices Files

To ensure that the correct org.apache.xml.security package that is required by Spring Framework is used, copy the jboss-deployment-structure.xml_forworkflow file and the jboss-deployment-structure.xml_forwipservices file to their appropriate directories and rename them, as follows:

For jboss-deployment-structure.xml_forworkflow:

1. Copy the jboss-deployment-structure.xml_forworkflow file from your *JBoss7Upgrade\jboss_deployment_structure* directory to the *JBoss7-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.workflow9.3.ear\META-INF* directory.
2. Change the name of the jboss-deployment-structure.xml_forworkflow file to jboss-deployment-structure.xml.

For jboss-deployment-structure.xml_forwipservices:

1. Copy the jboss-deployment-structure.xml_forwipservices file from the *JBoss7Upgrade\jboss_deployment_structure* directory to the *JBoss7-home directory\standalone\deployments\deploy_sas\SASServer1\sas.wip.services9.3.ear\META-INF* directory.
2. Then change the name of the jboss-deployment-structure.xml_forwipservices file to jboss-deployment-structure.xml.

Modify the JNDI Values in the sas.wip.soapservices.war File

The JNDI values in sas.wip.soapservices.war do not match the JNDI values that are defined in JBoss AS 7.1.1. Modify these values in sas.wip.soapservices.war so that they match the values that are defined in JBoss, as follows:

1. Locate the `jboss-web.xml` file that resides in `JBoss7-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.wip.services9.3.ear\sas.wip.soapservices.war\WEB-INF\`.
2. In `jboss-web.xml`, add the prefix `java:/` to all of the **JMS resource** `<jndi-name>` attributes, as shown in this example:

```
<jndi-name>java:/sas/jms/TopicConnectionFactory</jndi-name>
```

Remove the Unregistered Security Domain in JBoss AS 7.1.1

The security domain `java:/jaas/SASApplicationLogin` is not registered in JBoss AS 7.1.1. Therefore, you need to remove that security domain, as follows:

1. Locate the `jboss-web.xml` file that resides in the `JBoss7-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.wip.services9.3.ear\sas.biws.war\WEB-INF\` directory.
2. In `jboss-web.xml`, remove the following line:

```
<security-domain>java:/jaas/SASApplicationLogin</security-domain>
```

Modify the `sas.portlet.container.client.jar` File

You need to modify the contents of the `sas.portlet.container.client.jar` file, which resides in `JBoss-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.biportlets4.3.war\WEB-INF\lib`, as follows:

1. Create a directory named `update.jar` in any directory outside of `JBoss-home-directory`.
2. Copy `sas.portlet.container.client.jar` to the `update.jar` directory.
3. From a command window, navigate to the `update.jar` directory.
4. Submit the following command to extract the contents of the JAR file:

```
prompt: C:\jdk1.7.0_15\bin\jar xvf jar-name
```

Example: `C:\jdk1.7.0_15\bin\jar xvf sas.portlet.container.client.jar`

5. Delete the original copy of `sas.portlet.container.client.jar` from `JBoss-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.biportlets4.3.war\WEB-INF\lib`.
6. In the `update.jar` directory, change to the `META-INF` directory. Then create a subdirectory and name it `taglib`.
7. Move the `portlet-container-taglib.tld` from `META-INF` to `taglib`.
8. In the `portlet-container-taglib.tld` file, replace all occurrences of `<bodycontent>jsp</bodycontent>` with `<bodycontent>JSP</bodycontent>`. Then save `portlet-container-taglib.tld`.

- Submit the following command to combine the extracted files and directories:

```
prompt: C:\jdk1.7.0_15\bin\jar cvfm jar-name META-INF\MANIFEST.MF *
```

Example: C:\jdk1.7.0_15\bin\jar cvfm sas.portlet.container.client.jar
META-INF\MANIFEST.MF *

- Copy the newly created JAR file from the `update.jar` directory to the `lib` directory of your web application.

Sample Path for lib Directory: C:\jboss-as-7.1.1.Final\standalone\deployments\
deploy_sas\SASServer1\sas.biportlets4.3.ear\
sas.biportlets4.3.war\WEB-INF\lib

Modify Cookie Settings

Change the cookie settings in the web.xml files for the `sas.biportlets4.3.war`, `sas.jsr168remoteportlet4.3.war`, and `sas.portal.war` directories so that your web applications work consistently.

- Locate the web.xml file in the following directories:
 - `JBoss-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.biportlets4.3.war\WEB-INF\`
 - `JBoss-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.biportlets4.3.ear\sas.jsr168remoteportlet4.3.war\WEB-INF\`
 - `JBoss-home-directory\standalone\deployments\deploy_sas\SASServer1\sas.portal4.3.ear\sas.portal.war\WEB-INF\`
- In the web.xml file found in each of the directories above, make the following modifications and save them:
 - Change instances of the web.xml version in the `<web-app>` attribute 2.4 to 3.0, as shown below.

```
<web-app id="SASPortal" version="3.0"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-
    app_3_0.xsd">
```

- Add the `<cookie-config>` attribute to the `<session-config>` tag, as shown below:

```
<session-config>
  <session-timeout>30</session-timeout>
  <cookie-config>
    <path>/</path>
  </cookie-config>
</session-config>
```

Step 5: Start JBoss AS 7.1.1. and Validate SAS Web Applications

1. Start JBoss AS 7.1.1.

Note: In Microsoft Windows operating environments, you can start JBoss AS 7.1.1 as a Windows service because running `jbossScripting47.bat` replaced the previous version of JBoss in the Windows services.

2. Validate your SAS Enterprise BI Server web applications. For specific validation steps, see the `Instructions.html` file that resides in *SAS-configuration-directory*\Levn\Documents.

