

# **Roambi ES for SAS<sup>®</sup> 9.2 and 9.3 Installation, Configuration, and Deployment Guide**



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *Roambi ES for SAS® 9.2 and 9.3: Installation, Configuration, and Deployment Guide*. Cary, NC: SAS Institute Inc.

**Roambi ES for SAS® 9.2 and 9.3: Installation, Configuration, and Deployment Guide**

Copyright © 2011, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

**For a hardcopy book:** No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**For a Web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

**U.S. Government Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227–19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, July 2011

2nd printing, October 2011

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at

[support.sas.com/publishing](http://support.sas.com/publishing) or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

---

# Contents

<i>About This Document</i> . . . . .	<i>v</i>
<b>Chapter 1 • Overview</b> . . . . .	<b>1</b>
What Is Roambi ES for SAS? . . . . .	1
Prerequisites and Supporting Software . . . . .	1
Required Hardware and Operating Environments . . . . .	2
<b>Chapter 2 • Deploy Roambi ES for SAS</b> . . . . .	<b>5</b>
Install Roambi ES for SAS . . . . .	5
Configure Roambi ES for SAS . . . . .	8
Configuring the Java Memory Parameters on Application Servers . . . . .	10
Configuring the Application Server Connector . . . . .	13
Special Configuration Information for the Tomcat Application Server . . . . .	14
<b>Chapter 3 • Deploy Roambi ES for SAS to the JBoss Application Server</b> . . . . .	<b>15</b>
Deploying to the SAS Middle Tier Host . . . . .	15
Deploying to a Separate Host . . . . .	18
<b>Chapter 4 • Deploy Roambi ES for SAS to the WebLogic Server</b> . . . . .	<b>25</b>
Deploy Roambi ES to the WebLogic Server Version 10.3.0 . . . . .	25
<b>Index</b> . . . . .	<b>27</b>



# About This Document

---

## **Audience**

This document is a step-by-step guide for administrators who are installing Roambi Enterprise Server (ES) for SAS and configuring it to access the SAS Enterprise BI Server.



# Chapter 1

## Overview

---

<b>What Is Roambi ES for SAS?</b> .....	<b>1</b>
<b>Prerequisites and Supporting Software</b> .....	<b>1</b>
Roambi ES Software .....	1
SAS Software .....	2
<b>Required Hardware and Operating Environments</b> .....	<b>2</b>
Hardware Requirements .....	2
Supported Operating Environments .....	2
Mobile Device Requirements .....	2

---

## What Is Roambi ES for SAS?

Roambi ES for SAS integrates Roambi ES3 with the SAS Enterprise BI Server. Roambi ES for SAS transforms SAS Web Report Studio reports into Roambi (RBI) files, which can be downloaded to compatible mobile devices.

This document provides a list of steps to configure the Roambi ES for SAS deployment. It addresses deploying Roambi ES for SAS with the JBoss Application Server, the Oracle WebLogic Server, or the Tomcat Application Server. For troubleshooting information, see the *Roambi ES3 Configuration and Installation Guide* that is provided by MeLLmo and is installed with Roambi ES.

For information about using Roambi ES for SAS on mobile devices, see the *Roambi ES for SAS: User's Guide*, which is available from [support.sas.com/documentation/cdl/en/citmblug/64712/PDF/default/citmblug.pdf](http://support.sas.com/documentation/cdl/en/citmblug/64712/PDF/default/citmblug.pdf).

---

## Prerequisites and Supporting Software

### *Roambi ES Software*

Roambi ES is supplied as a self-extracting archive file. When opened, the archive file unpacks files into a directory. Configuration is a manual process after the files are unpacked. The installed files include the *Roambi ES3 Configuration and Installation Guide* and other Roambi documentation.

Roambi ES is a small Web application requiring a J2EE server; it's distributed as a WAR file. A Roambi ES license file is needed to activate the software, which is

deployed to the J2EE server with the application. The application fails to start if the license file is missing.

No additional adapter or connector is required for Roambi ES to connect to SAS; functionality has been built into the Roambi software. After it's installed, Roambi ES software only requires connection information to enable access to a SAS Enterprise BI Server. Roambi ES connects to a SAS 9.2 or SAS 9.3 Enterprise BI Server through a Web service extension to SAS Web Report Studio 4.3 or 4.31. Two SAS hot fixes extend SAS Web Report Studio 4.3 and add the Web service.

## **SAS Software**

Roambi ES requires the SAS 9.2 Enterprise BI Server and SAS Web Report Studio 4.3 with the required hot fixes applied or the SAS 9.3 Enterprise BI Server and SAS Web Report Studio 4.31. The following URL returns an XML description of the Roambi File Web Service (WSDL): `http://<hostname>:<port>/SASWebReportStudio/services/RoambiFileWebService`.

---

# **Required Hardware and Operating Environments**

## **Hardware Requirements**

Roambi ES for SAS has the following minimum hardware requirements for Roambi ES:

- 2.0 GHz Pentium 4-class processor
- 4.0 GB RAM
- 2 GB of disk storage

## **Supported Operating Environments**

Roambi ES for SAS software is supported only on the following platforms for Roambi ES:

- Windows Server 2003 and 2008
- Mac OS X
- Red Hat Enterprise Linux 5.3 and 5.4
- SUSE 10

## **Mobile Device Requirements**

### **Supported Mobile Devices**

The Roambi ES for SAS client application is supported only on the following devices:

- Apple iPhone 3G, 3GS, or 4 with either iOS 3.x or 4.x
- Apple iPad with either iOS 3.2 or 4.2
- Apple iPad 2 with iOS 4.2
- Apple iPod Touch with either iOS 3.x or 4.x

- BlackBerry Bold Series (9000, 9700, or 9780) or BlackBerry Torch 9800 with either the BlackBerry Device Software 5.0 or the BlackBerry 6 OS. (The BlackBerry 6 OS is recommended.)

### **Prerequisites for BlackBerry Devices**

The requirements for Blackberry devices are as follows:

- The Roambi reports are stored on a MicroSD card. **Mass Storage Mode** must be turned off to run the Roambi Visualizer.
- **Application Control Permissions** must be granted to run the Roambi client application, specific Roambi features, or both.
- The **Access to the Interprocess Communication API** permission is required to persist, back up, and restore application settings such as saved portals.
- The **Access to the File API** permission is required to save and retrieve Roambi reports on a BlackBerry device's MicroSD card.



## Chapter 2

# Deploy Roambi ES for SAS

---

<b>Install Roambi ES for SAS</b> .....	<b>5</b>
Installing on Windows .....	5
Installing on Linux .....	7
<b>Configure Roambi ES for SAS</b> .....	<b>8</b>
Step 1: Extract the Roambi ES Application .....	8
Step 2: Configure the Roambi ES Connection to SAS .....	9
Step 3: Install the Roambi ES License File .....	10
Step 4: (Optional) Configure the SMTP Server for Roambi ES .....	10
<b>Configuring the Java Memory Parameters on Application Servers</b> .....	<b>10</b>
Java Heap Size (-Xms and -Xmx) .....	10
Java Permanent Generation Size (-XX:PermSize and -XX:MaxPermSize) .....	11
JBoss 4.x on Windows or Linux .....	11
Tomcat 5.x and 6.x on Windows .....	11
Tomcat 5.x and 6.x on Linux .....	11
WebLogic 10.3.x on Windows .....	11
WebLogic 10.3.x on Linux .....	12
<b>Configuring the Application Server Connector</b> .....	<b>13</b>
Connector Max Post Size (maxPostSize) .....	13
JBoss 4.x on Windows or Linux .....	13
Tomcat 5.x and 6.x on Windows or Linux .....	13
WebLogic 10.3.x on Windows or Linux .....	14
<b>Special Configuration Information for the Tomcat Application Server</b> .....	<b>14</b>

---

## Install Roambi ES for SAS

### *Installing on Windows*

To install Roambi ES on Windows:

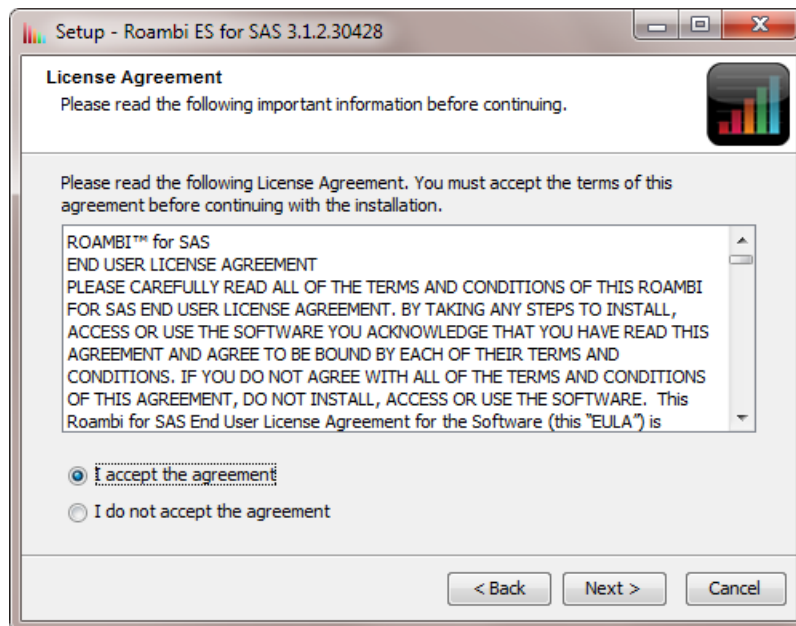
1. Launch the Roambi installer to open the Setup Wizard. Click **Next**.

**Display 2.1** Roambi ES for SAS Setup Wizard – Welcome Page

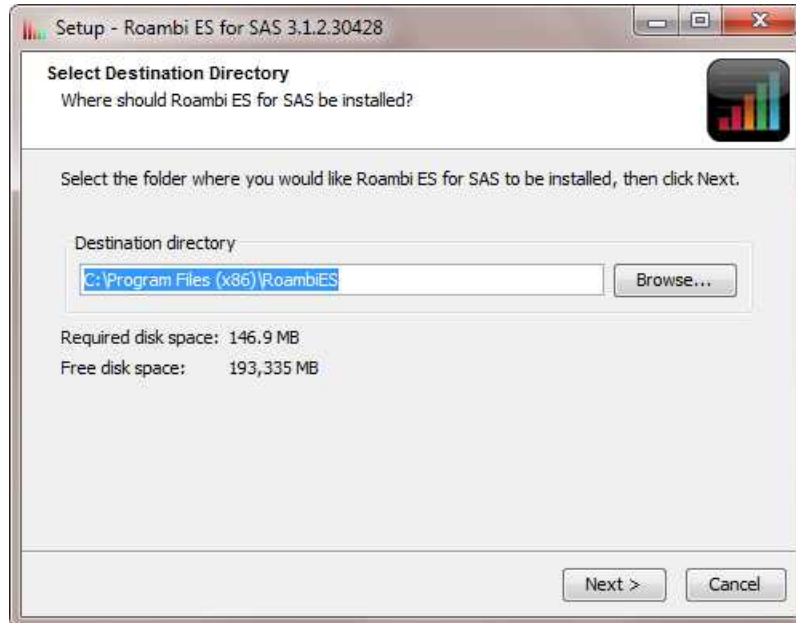


2. Accept the license agreement and click **Next**.

**Display 2.2** Roambi ES for SAS Setup Wizard – License Agreement Page



3. Accept the default installation directory or click **Browse** to specify a new installation directory. Click **Next**.

**Display 2.3** Roambi ES for SAS Setup Wizard – Select Destination Directory Page

The installer extracts the roambi.war file.

4. Click **Finish** to complete the installation and close the wizard.

**Display 2.4** Roambi ES for SAS Setup Wizard – Complete Page**Installing on Linux**

Run the `RoambiES_3_1_(version)_unix_1.0.sh` installer to extract the roambi.war file.

## Configure Roambi ES for SAS

The Roambi ES application is distributed as a WAR file. Configuration of the application involves editing files within the archive. Therefore, the archive is extracted and some files are edited.

### Step 1: Extract the Roambi ES Application

To extract the archive:

1. Create a new directory for the extracted archive:

```
<Install_Path>/roambi.war
```

2. Copy the roambi.war file into the new **roambi.war** directory.

3. Extract the roambi.war file using the following Java code on the command line from within the **roambi.war** directory:

```
<JAVA_Home>/bin/jar xvf roambi.war
```

4. Delete the roambi.war file from within the **<Install\_Path>/roambi.war** directory.

This completes the extraction of the Roambi ES application and results in the file structure shown here:

**Display 2.5** Contents of the roambi.war File

Name	Type	Modified	Size
css	Folder	4/14/2011 4:12 PM	
designer	Folder	4/14/2011 4:12 PM	
images	Folder	4/14/2011 4:12 PM	
img	Folder	4/14/2011 4:12 PM	
META-INF	Folder	4/14/2011 4:12 PM	
scripts	Folder	4/14/2011 4:12 PM	
WEB-INF	Folder	4/14/2011 4:12 PM	
check_network.jsp	Active Server...	3/24/2011 6:35 PM	3,791
error.css	Cascading St...	2/26/2011 8:34 PM	3,948
error.jsp	Active Server...	2/26/2011 8:34 PM	3,257
index.jsp	Active Server...	3/11/2011 4:21 PM	2,826
launch_bb.jsp	Active Server...	3/4/2011 5:07 PM	249
main.css	Cascading St...	2/26/2011 8:34 PM	1,885
status.jsp	Active Server...	2/26/2011 8:34 PM	4,551
swfEmbed.htm	HTML Docu...	2/26/2011 8:34 PM	1,773

## Step 2: Configure the Roambi ES Connection to SAS

Roambi ES must be configured to connect to the SAS Web Report Studio Web service to display SAS reports. To configure the connection:

1. Edit the `<Install_Path>/roambi.war/WEB-INF/roambi-settings.xml` file.

*Note:* Use a plain text editor to modify the settings file.

2. The following content is required to make the connection. Make sure that the `<repository>` entry is not commented out:

```
<settings>
<!--
*****
* Template: SAS *
*****
-->
<repository>
  <name>SAS</name>
  <id>sas</id>
  <type>com.mellmo.roambi.portals.plugin.sas.SASContentSourceFactory</type>
  <description>SAS Server</description>
  <params>
    <param name="service_url">SERVICE_URL_HERE</param>
  </params>
</repository>
</settings>
```

In the settings.xml file, the `<id>` must be unique. For example, if you have development, test, and production environments, you might want to use **sasdev**, **sasdevtest**, and **sasprod** as identifiers. Do not modify the `<type>` setting for Roambi ES.

3. Replace the **SERVICE\_URL\_HERE** text with the fully qualified URL for SAS Web Report Studio (for example, `http://<hostname>:<port>/SASWebReportStudio`).

*Note:* The URL is case-sensitive.

Additional repository entries can be appended for connections to other SAS Web Report Studio instances (for example, if you use development, test, and production environments).

Here's an example of an XML file with two connections in it:

```
<settings>
<!--
*****
* Template: SAS *
*****
-->
<repository>
  <name>SAS9.2</name>
  <id>Dev</id>
  <type>com.mellmo.roambi.portals.plugin.sas.SASContentSourceFactory</type>
  <description>SAS Portal</description>
  <params>
```

```

        <param name="service_url">xxx.xxx.xxx.xxx:8080/SASWebReportStudio</param>
    </params>
</repository>
<repository>
    <name>SAS9.3</name>
    <id>Prod</id>
    <type>com.mellmo.roambi.portals.plugin.sas.SASContentSourceFactory</type>
    <description>SAS Portal</description>
    <params>
        <param name="service_url">xxx.xxx.xxx.xxx:8080/SASWebReportStudio</param>
    </params>
</repository>
</settings>

```

### Step 3: Install the Roambi ES License File

To install the Roambi ES license file, copy the `roambi.licensekey` file to the `<Install_Path>/Roambi.war/WEB-INF/classes` directory.

### Step 4: (Optional) Configure the SMTP Server for Roambi ES

To send mail from Roambi ES for SAS, SMTP (Simple Mail Transfer Protocol) must be set up and running on an accessible server. To configure the server to send mail from Roambi ES:

1. Edit the `<Install_Path>/roambi.war/WEB-INF/classes/com/mellmo/roambi/util/mail.properties` file.
2. Set the host and other details of the SMTP server. In the following example, `localhost` is used. Typically, the SMTP will be a corporate mail server gateway.

```

protocol = smtp
host = localhost
port = 25
corePoolSize = 3
#username = username
#password = password
#rootURL = https://xxx.xxx.xxx/xxx

```

---

## Configuring the Java Memory Parameters on Application Servers

Configuration of individual application servers varies. You need to change the Java memory parameters to properly run Roambi ES.

### Java Heap Size (-Xms and -Xmx)

These settings control the size of the Java heap. Tuning this parameter properly reduces the overhead of garbage collection, which improves both the Roambi ES response time and the throughput.

## Java Permanent Generation Size (-XX:PermSize and -XX:MaxPermSize)

The section of the heap that is reserved for the permanent generation holds all of the reflective data for the JVM. Increase this size to optimize the performance of Roambi ES, which dynamically loads and unloads many classes.

### JBoss 4.x on Windows or Linux

Add the following entry to the run.conf file in the JBoss bin folder. Create the file if it doesn't exist.

```
JAVA_OPTS="-Xms1024m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=256m"
```

### Tomcat 5.x and 6.x on Windows

Open the Tomcat Configuration Utility and select the **Java** tab. Assign the **Initial memory pool** and **Maximum memory pool** to **1024**. Add the following lines to **Java Options**:

```
-XX:MaxPermSize=256m
-XX:PermSize=256m
```

### Tomcat 5.x and 6.x on Linux

Add the following entry to the setenv.sh file in the Tomcat **bin** folder. Create the file if it doesn't exist.

```
export JAVA_OPTS="-Xms1024m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=256m"
```

### WebLogic 10.3.x on Windows

Edit the setDomainEnv.cmd file in the WebLogic domain **bin** folder. Make the following updates:

1. Search for WLS\_MEM\_ARGS\_64BIT. You should find something like this:

```
set WLS_MEM_ARGS_64BIT = -Xms256m -Xmx512m
```

Replace both occurrences with this:

```
set WLS_MEM_ARGS_64BIT = -Xms1024m -Xmx1024m
```

2. Search for WLS\_MEM\_ARGS\_32BIT. You should find something like this:

```
set WLS_MEM_ARGS_32BIT = -Xms256m -Xmx512m
```

Replace both occurrences with this:

```
set WLS_MEM_ARGS_32BIT = -Xms1024m -Xmx1024m
```

3. Search for MEM\_PERM\_SIZE\_64BIT. You should find something like this:

```
set MEM_PERM_SIZE_64BIT = -XX:PermSize=128m
```

Replace all occurrences with this:

```
set MEM_PERM_SIZE_64BIT = -XX:PermSize=256m
```

4. Search for MEM\_PERM\_SIZE\_32BIT. You should find something like this:

```
set MEM_PERM_SIZE_32BIT = -XX:PermSize=48m
```

Replace all occurrences with this:

```
set MEM_PERM_SIZE_32BIT = -XX:PermSize=256m
```

5. Search for MEM\_\_MAX\_PERM\_SIZE\_64BIT. You should find something like this:

```
set MEM_MAX_PERM_SIZE_64BIT = -XX:PermSize=128m
```

Replace all occurrences with this:

```
set MEM_MAX_PERM_SIZE_64BIT = -XX:PermSize=256m
```

6. Search for MEM\_MAX\_PERM\_SIZE\_32BIT. You should find something like this:

```
set MEM_MAX_PERM_SIZE_32BIT = -XX:PermSize=128m
```

Replace all occurrences with this:

```
set MEM_MAX_PERM_SIZE_32BIT = -XX:PermSize=256m
```

### WebLogic 10.3.x on Linux

Edit the setDomainEnv.sh file in the WebLogic domain **bin** folder. Make the following updates:

1. Search for WLS\_MEM\_ARGS\_64BIT. You should find something like this:

```
WLS_MEM_ARGS_64BIT = "-Xms256m -Xmx512m"
```

Replace both occurrences with this:

```
WLS_MEM_ARGS_64BIT = "-Xms1024m -Xmx1024m"
```

2. Search for WLS\_MEM\_ARGS\_32BIT. You should find something like this:

```
WLS_MEM_ARGS_32BIT = "-Xms256m -Xmx512m"
```

Replace both occurrences with this:

```
WLS_MEM_ARGS_32BIT = "-Xms1024m -Xmx1024m"
```

3. Search for MEM\_PERM\_SIZE\_64BIT. You should find something like this:

```
MEM_PERM_SIZE_64BIT = "-XX:PermSize=128m"
```

Replace all occurrences with this:

```
MEM_PERM_SIZE_64BIT = "-XX:PermSize=256m"
```

4. Search for MEM\_PERM\_SIZE\_32BIT. You should find something like this:

```
MEM_PERM_SIZE_32BIT = "-XX:PermSize=48m"
```

Replace all occurrences with this:

```
MEM_PERM_SIZE_32BIT = "-XX:PermSize=256m"
```

5. Search for MEM\_\_MAX\_PERM\_SIZE\_64BIT. You should find something like this:

```
MEM_MAX_PERM_SIZE_64BIT = "-XX:PermSize=128m"
```

Replace all occurrences with this:

```
MEM_MAX_PERM_SIZE_64BIT = "-XX:PermSize=256m"
```

6. Search for `MEM_MAX_PERM_SIZE_32BIT`. You should find something like this:

```
MEM_MAX_PERM_SIZE_32BIT = "-XX:PermSize=128m"
```

Replace all occurrences with this:

```
MEM_MAX_PERM_SIZE_32BIT = "-XX:PermSize=256m"
```

## Configuring the Application Server Connector

### Connector Max Post Size (`maxPostSize`)

The `maxPostSize` setting increases the maximum POST size. The POST request maximum size is in bytes, which is handled by the container FORM URL parameter parsing. This setting is useful when publishing RBI files that are based on reports with a large amount of columns and data. A value of `"-1"` disables the feature, but this is not recommended because it can make the site vulnerable to DDoS attacks if the server is accessible from the Internet.

### JBoss 4.x on Windows or Linux

Edit the `server.xml` file in the `<JBoss deploy>/jboss-web.deployer` folder and search for the `<connector>` tag. Add the `maxPostSize=6291456` property.

A connector tag looks like this:

```
<Connector
  port="8080"
  address="{jboss.bind.address}"
  maxThreads="250"
  maxHttpHeaderSize="8192"
  emptySessionPath="true" protocol="HTTP/1.1"
  enableLookups="false"
  redirectPort="8443"
  acceptCount="100"
  connectionTimeout="20000"
  disableUploadTimeout="true"
  maxPostSize=6291456
/>
```

### Tomcat 5.x and 6.x on Windows or Linux

Edit the `server.xml` file in the Tomcat `conf` folder and search for the `<connector>` tag. Add the `maxPostSize=6291456` property.

A connector tag looks like this:

```
<Connector
  port="8080"
  protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  maxPostSize=6291456
/>
```

**WebLogic 10.3.x on Windows or Linux**

Open the WebLogic Server Administration Console and select the server or virtual server where Roambi ES is deployed. Select the **Protocols** tab and then the **HTTP** tab. For the **MaxPostSize** field, change the value to **6291456**.

---

## Special Configuration Information for the Tomcat Application Server

When running SSL, the Tomcat application server is configured to use a secure connector on port **443**, which references a certificate store. In this configuration, Tomcat does not actually run the SSL protocol, but it sends proxy requests to the BIG-IP virtual server in a secured format using the HTTP protocol. The advantage of this configuration is that SSL needs to be installed on the BIG-IP virtual server, not on each Tomcat in the server farm (which would be necessary otherwise).

Modify the default HTTP connector in the `<Tomcat_home>/conf/server.xml` file:

1. Change the default port from `"8080"` to `"886"`.
2. Add the argument `"maxPostSize="-1"`.
3. Add the arguments for `"secure="true"` and `"proxyPort="443"`.

The Tomcat connector should look like this:

```
<Connector
  port="886"
  protocol="HTTP/1.1"
  maxPostSize="-1"
  connectionTimeout="20000"
  secure="true"
  proxyPort="443"
/>
```

This connector definition sends proxies for any Tomcat port **80** to port **443** on the BIG-IP virtual server. In this case, port **443** (the default SSL on the BIG-IP virtual server) is redirected to the designated port **886** for Tomcat HTTP. Because Tomcat is not running SSL, it is necessary to define this traffic as secure, as shown previously. You cannot access Tomcat without first going through the BIG-IP virtual server.

A second connector should be added for direct access to Tomcat through HTTP for testing purposes. In the following example, port **880** is chosen as a nonstandard port, as opposed to the default port **8080**:

```
<Connector
  port="880"
  protocol="HTTP/1.1"
  maxPostSize="-1"
  connectionTimeout="20000"
/>
```

## Chapter 3

# Deploy Roambi ES for SAS to the JBoss Application Server

---

<b>Deploying to the SAS Middle Tier Host</b> .....	<b>15</b>
Prepare the Roambi ES JBoss Server Instance .....	15
Set the JVM Options for the Roambi ES Instance .....	17
Deploy Roambi ES to JBoss .....	17
Creating a Windows Service .....	17
Starting and Stopping the Roambi ES JBoss Instance .....	17
<b>Deploying to a Separate Host</b> .....	<b>18</b>
Prepare the Roambi ES JBoss Server Instance .....	18
Deploy Roambi ES to the JBoss Application Server .....	24
Creating a Windows Service .....	24
Starting and Stopping the Roambi ES JBoss Instance .....	24

---

## Deploying to the SAS Middle Tier Host

This approach deploys Roambi ES on the host where the SAS middle tier is already deployed. The JBoss server instance that is configured for SAS is reused to create a new JBoss server instance. A new server instance is used so that Roambi can be restarted without restarting the entire SAS middle tier.

### *Prepare the Roambi ES JBoss Server Instance*

A new Roambi ES JBoss server instance is created. The existing **SASServer1** is used as the basis for the new Roambi ES JBoss server instance. It is important at this point to stop **SASServer1** and to make sure that the Java process has stopped completely before continuing.

*Note:* The JBoss EAP 4.3 has library conflicts with Roambi ES. The following JAR files must be removed from the Roambi deployment folder (**WEB-APPS/roambi/lib**):

- **xml-apis-1.0.b2.jar**
- **stax-api-1.0.1.jar**
- **stax-api-1.0.2.jar**
- **stax-1.2.0.jar**

To prepare the Roambi ES instance of JBoss:

1. Copy the **<JBoss\_Home>/server/SASServer1** directory to **<JBoss\_Home>/server/RoambiES**.

2. In the new directory, delete the contents of the following directories:
  - `<JBoss_Home>/server/RoambiES/deploy_sas`
  - `<JBoss_Home>/server/RoambiES/tmp`
  - `<JBoss_Home>/server/RoambiES/work`
3. Delete the following files:
  - `<JBoss_Home>/server/RoambiES/deploy/sas-mail-service.xml`
  - `<JBoss_Home>/server/RoambiES/deploy/SharedServices-ds.xml`
  - `<JBoss_Home>/server/RoambiES/deploy/jms/sas-jms-service.xml`
4. Edit the `<JBoss_Home>/server/RoambiES/conf/jboss-service.xml` file and replace all instances of `SASServer1` with `RoambiES`.
5. Edit the `<JBoss_Home>/server/RoambiES/deploy/jboss-web.deployer/server.xml` file and add the following option to the Connector with port 8080: `maxPostSize="6291456"`.

*Note:* 6291456 is the maximum size of the HTTP post in bytes. It limits the maximum RBI file size that can be transferred to 6 MB. For very large RBI files, this value can be increased. This is the recommended default setting. For more information, see the *Roambi ES3 Installation and Configuration Guide*.

6. Edit the `<JBoss_Home>/server/RoambiES/jboss-port-bindings.xml` file. Find the `ports-01` server element and rename the server element name attribute from `ports-01` to `RoambiES`.

This file contains definitions for the sets of ports to use for a JBoss server instance, such as `ports-01` or `ports-02`. This step identifies a free port set and renames the set to `RoambiES`.

Change `<server name="ports-01">` to `<server name="RoambiES">`.

If the SAS Deployment Wizard has deployed `SASServer1` and `SASServer2`, then the port set `ports-01` has already been used for `SASServer2`, and `ports-01` has been renamed as `SASServer2` and is not available. In this case, rename `<server name="ports-02">` to `<server name="RoambiES">` instead. If `<server name="ports-02">` is not available, then use `<server name="ports-03">`, and so on, until you find a free port.

7. Copy the `<JBoss_Home>/bin/SASServer1.bat (.sh)` file to `<JBoss_Home>/bin/RoambiES.bat (.sh)`.
8. Edit the `<JBoss_Home>/bin/RoambiES.bat (.sh)` file and replace all instances of `SASServer1` with `RoambiES`.
9. For Windows only, edit the `<JBoss_Home>/server/RoambiES/wrapper.conf` file. Make the following changes:
  - Replace all instances of `SASServer1` with `RoambiES`.
  - Comment out the following line by putting a number sign (#) at the start of this line:
 

```
wrapper.ntservice.dependency.1=SAS [config-Lev1] Remote Services
```
  - Update the log file by changing `wrapper.log` to `wrapper-roambi.log` in the following: `wrapper.logfile=../logs/wrapper-roambi.log`.

## Set the JVM Options for the Roambi ES Instance

The Roambi ES instance of JBoss does not require the same JVM options as the SAS instance of JBoss.

To set the correct JVM options for Windows:

1. Edit the `<JBoss_Home>/bin/Roambi.bat` file and replace the existing `set JAVA_OPTS` line with the following:

```
set JAVA_OPTS=-Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256
```

2. Edit the `<JBoss_Home>/server/RoambiES/wrapper.conf` file. Remove all of the `wrapper.java.additional` lines and replace them with the following:

```
wrapper.java.additional.1 = -Xms2048m
wrapper.java.additional.2 = -Xmx2048m
wrapper.java.additional.3 = -XX:PermSize=256m
wrapper.java.additional.4 = -XX:MaxPermSize=256m
```

For Linux, edit the `<JBoss_Home>/bin/Roambi.sh` file and replace the existing `set JAVA_OPTS` line with the following:

```
JAVA_OPTS="-Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256"
```

## Deploy Roambi ES to JBoss

To deploy the Roambi ES application, copy the `<Install_Path>/Roambi.war` file to the `<JBoss_Home>/server/RoambiES/deploy_sas` directory.

## Creating a Windows Service

For Windows, a service can be created for the new instance of JBoss. Run the following code from a command prompt:

```
<JBoss_Home>/bin/RoambiES.bat install
```

## Starting and Stopping the Roambi ES JBoss Instance

The Roambi ES JBoss instance is now configured and can be started. For Windows, it can be started as a Windows service or by using this script:

```
<JBoss_Home>/bin/RoambiES.bat start
```

For Linux, it can be started by using this script:

```
<JBoss_Home>/bin/RoambiES.sh start
```

The Roambi Designer application is now accessible on `http://<hostname>:<port>/roambi/designer`.

If you used the port set `ports-01`, then the port is **8180**. If you used the port set `ports-02`, then the port is **8280**, and so on.

## Deploying to a Separate Host

The following approach deploys Roambi ES on a host without SAS software. The JBoss instance needs to be configured from the initial installation of JBoss.

### Prepare the Roambi ES JBoss Server Instance

To prepare the Roambi ES instance of the JBoss server:

1. Install or extract JBoss.
2. Copy the `<JBoss_Home>/server/default` directory to `<JBoss_Home>/server/RoambiES`.
3. Edit the `<JBoss_Home>/server/RoambiES/conf/jboss-service.xml` file and search for `ServiceBindingManager`. Uncomment the section and set `ServerName` to `ports-default`, as shown here:

```
<mbean code="org.jboss.services.binding.ServiceBindingManager"
      name="jboss.system:service=ServiceBindingManager">
  <attribute name="ServerName">ports-default</attribute>
  <attribute name="StoreURL">${jboss.home.url}/docs/examples/binding-
manager/sample-bindings.xml</attribute>
  <attribute name="StoreFactoryClassName">
    org.jboss.services.binding.XMLServicesStoreFactory
  </attribute>
</mbean>
```

4. Edit the `<JBoss_Home>/server/RoambiES/deploy/jboss-web.deployer/server.xml` file and add the following option to the Connector with port 8080:

```
maxPostSize="6291456"
```

For a Windows deployment, create the required scripts to start and stop the Roambi ES JBoss instance:

1. Edit the `<JBoss_Home>/bin/run.bat` file and replace the line

```
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m
```

with

```
set JAVA_OPTS=%JAVA_OPTS%
```

2. Create the `<JBoss_Home>/bin/RoambiES.bat` file with the following content:

*Note:* You must update the paths within the script (for example, `<<JBOSS_HOME>>` or `<<JDK_PATH>>`).

```
@echo off
rem -----
rem JBoss Bootstrap Script for Win32
rem -----

setlocal

set SERVICENAME=JBoss - RoambiES
set JBOSS_BIN_DIR=<<JBOSS_HOME>>/bin
```

```

set JBOSS_SERVER_DIR=<<JBOSS_HOME>>/server/RoambiES
set WRAPPER_DIR=<<JBOSS_HOME>>/service

if x%1 EQU x goto start_as_script
if %1 EQU install goto install
if %1 EQU remove goto remove
if %1 EQU start goto start
if %1 EQU stop goto stop
if %1 EQU restart goto restart
if %1 EQU pause goto pause
if %1 EQU resume goto resume
if %1 EQU status goto status
if %1 EQU -install goto install
if %1 EQU -remove goto remove
if %1 EQU -start goto start
if %1 EQU -stop goto stop
if %1 EQU -restart goto restart
if %1 EQU -pause goto pause
if %1 EQU -resume goto resume
if %1 EQU -status goto status
goto usage

:install start /b /wait "JBoss - RoambiES" "%WRAPPER_DIR%\Wrapper.exe" -i "%JBOSS_SERVER_DIR%\wrapper.conf"
    if %ERRORLEVEL% == 0 echo JBoss server installed as a service
    goto end
:remove
    start /b /wait "JBoss - RoambiES" "%WRAPPER_DIR%\Wrapper.exe" -r "%JBOSS_SERVER_DIR%\wrapper.conf"
    if %ERRORLEVEL% == 0 echo JBoss server removed as a service
    goto end
:start
    net start "%SERVICENAME%"
    if %ERRORLEVEL% == 0 echo JBoss server started as a service
    goto end
:stop
    net stop "%SERVICENAME%"
    if %ERRORLEVEL% == 0 echo JBoss server stopped as a service
    goto end
:restart
    net stop "%SERVICENAME%"
    if %ERRORLEVEL% == 0 echo JBoss server stopped as a service
    net start "%SERVICENAME%"
    if %ERRORLEVEL% == 0 echo JBoss server started as a service
    goto end
:pause
    net pause "%SERVICENAME%"
    if %ERRORLEVEL% == 0 echo JBoss server paused as a service
    goto end
:resume
    net continue "%SERVICENAME%"
    if %ERRORLEVEL% == 0 echo JBoss server resumed as a service
    goto end
:status
    sc interrogate "%SERVICENAME%"
    goto end
:usage
    echo USAGE: %0 {install ^| start ^| stop ^| restart ^| pause ^| resume ^| remove ^| status}

```

```

goto end

:start_as_script
set JAVA_HOME=<<JDK_PATH>>
set JAVA_OPTS=-Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256

"C:\SAS9.2\thirdparty\jboss-4.2.3.GA\bin\run.bat" -c RoambiES -b 0.0.0.0 %*

:end
endlocal
if [%2] EQU [exit] exit %ERRORLEVEL%

```

3. Create the `<JBoss_Home>/server/RoambiES/wrapper.conf` file with the following content. For more information about formats and log levels, see the *Roambi ES3 Configuration and Installation Guide*.

*Note:* You must update the paths within the script (for example, `<<JBOSS_HOME>>` or `<<JDK_PATH>>`).

```

*****
# Wrapper Properties
*****
# Java Application

wrapper.java.command=<<JDK_PATH>>\bin\java

# The Java Main class. This class must implement the WrapperListener
# interface or guarantee that the WrapperManager class is initialized.
# Helper classes are provided to do this for you.
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp

# Java Classpath (include wrapper.jar)
# Add class path elements as needed starting from 1.
wrapper.java.classpath.1=<<JBOSS_HOME>>/service/wrapper.jar
wrapper.java.classpath.2=<<JDK_PATH>>/lib/tools.jar
wrapper.java.classpath.3=<<JBOSS_HOME>>/bin/run.jar

# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path.1=<<JBOSS_HOME>>/service

# Java Additional Parameters
wrapper.java.additional.1=-Xms2048m
wrapper.java.additional.2=-Xmx2048m
wrapper.java.additional.3=-XX:PermSize=256m
wrapper.java.additional.4=-XX:MaxPermSize=256m
# Java Additional Parameters End

# Application Parameters
# Add the parameters as needed starting from 1.
wrapper.app.parameter.1=org.jboss.Main
wrapper.app.parameter.2=-c
wrapper.app.parameter.3=RoambiES
wrapper.app.parameter.4=-b

wrapper.app.parameter.5=0.0.0.0

*****

```

```

# Wrapper Logging Properties
#*****
# The format of the output for the console.
wrapper.console.format=PM

# The log level for console output.
wrapper.console.loglevel=WARN

# The log file to use for wrapper output logging.
wrapper.logfile=../logs/wrapper-roambi.log

# The format of output for the log file.
wrapper.logfile.format=LPTM

# The level for the output in the log file.
wrapper.logfile.loglevel=WARN

# The maximum size that the log file will be allowed to grow to before
# the log is rolled. The size is specified in bytes. The default value
# of 0 disables log rolling. You can abbreviate with the 'k' (kb)
# or 'm' (mb) suffix. For example: 10m = 10 megabytes.
wrapper.logfile.maxsize=0

# The maximum number of rolled log files that are allowed before the old
# files are deleted. The default value of 0 implies no limit.
wrapper.logfile.maxfiles=0

# The log level for sys/event log output.
wrapper.syslog.loglevel=ERROR

#*****
# Wrapper Windows Properties
#*****
# The title to use when running as a console.
wrapper.console.title="JBoss - RoambiES"

#*****
# Wrapper Windows NT/2000/XP Service Properties
#*****
# WARNING - Do not modify any of these properties when an application
# using this configuration file has been installed as a service.
# Uninstall the service before modifying this section.
# The service can then be reinstalled.

# The name of the service.
wrapper.ntservice.name=JBoss - RoambiES

# The display name of the service.
wrapper.ntservice.displayname=JBoss - RoambiES

# A description of the service.
wrapper.ntservice.description=JBoss - RoambiES

# The mode in which the service is installed. (AUTO_START or DEMAND_START)
wrapper.ntservice.starttype=DEMAND_START

```

```
# Allow the service to interact with the desktop?
wrapper.ntservice.interactive=false

# Wait 5 minutes for the JVM to shut itself down before terminating.
wrapper.shutdown.timeout=300
wrapper.jvm_exit.timeout=300
```

For a Linux deployment, create this script to start and stop the Roambi ES JBoss instance:

*Note:* You must update the paths within the script (for example, <<JBOSS\_HOME>> or <<JDK\_PATH>>).

```
#!/bin/sh -p
### ===== ###
## ##
## JBoss Bootstrap Script ##
## ##
### ===== ###

JAVA_HOME="<<JDK_PATH>>"
export JAVA_HOME

JAVA_OPTS="-Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256"
export JAVA_OPTS

mkdir -p "<<JBOSS_HOME>>/server/RoambiES/log"

# Get argument
if [ "$1" = "-start" ]; then
    arg=start
elif [ "$1" = "-stop" ]; then
    arg=stop
elif [ "$1" = "-status" ]; then
    arg=status
elif [ "$1" = "-restart" ]; then
    arg=restart
elif [ "$1" = "-kill" ]; then
    arg=kill
elif [ "$1" = "start" ]; then
    arg=start
elif [ "$1" = "stop" ]; then
    arg=stop
elif [ "$1" = "status" ]; then
    arg=status
elif [ "$1" = "restart" ]; then
    arg=restart
elif [ "$1" = "kill" ]; then
    arg=kill
else
    arg=$1
fi

shift

case "$arg" in
    start)
```

```

    LAUNCH_JBOSS_IN_BACKGROUND=true
    export LAUNCH_JBOSS_IN_BACKGROUND
    nohup "<<JBOSS_HOME>>/bin/run.sh" -c RoambiES -b 0.0.0.0 $@ >
        "<<JBOSS_HOME>>/server/RoambiES/log/stdout.log" 2>&1 &
    pid=$!
    echo $pid > "<<JBOSS_HOME>>/server/RoambiES/log/jboss.pid"
    echo "JBoss startup has begun (pid $pid)..."
;;

stop)
# Add args for username/password as in "RoambiES.sh stop -u username -p password"
    JAVA_OPTS="-d64 -Xms512M -Xmx512M"
    "<<JBOSS_HOME>>/bin/shutdown.sh" -s localhost:1099 $* -S
;;

kill)
    if [ -f <<JBOSS_HOME>>/server/RoambiES/log/jboss.pid ]; then
        pid=`cat <<JBOSS_HOME>>/server/RoambiES/log/jboss.pid`
        kill $* $pid
        if [ $? -ne 0 ]; then
            echo "pid: $pid"
        fi
    else
        echo "JBoss is not running"
        exit 1
    fi
;;

status)
    if [ -f <<JBOSS_HOME>>/server/RoambiES/log/jboss.pid ]; then
        pid=`cat <<JBOSS_HOME>>/server/RoambiES/log/jboss.pid`
        kill -0 $pid > /dev/null 2>&1
        if [ $? -eq 0 ]; then
            echo "JBoss running (pid $pid)"
        else
            echo "JBoss is not running"
        fi
    else
        echo "JBoss is not running"
    fi
;;

restart)
    $0 -kill -9
    if [ $? -eq 0 ]; then
        sleep 10
        $0 -start
    fi
;;

*)
    "<<JBOSS_HOME>>/bin/run.sh" -c RoambiES -b 0.0.0.0 $@
;;
esac

```

```
exit 0
```

```
exit 0
```

### **Deploy Roambi ES to the JBoss Application Server**

To deploy the Roambi ES application, copy the `<Install_Path>/Roambi.war` directory to `<JBoss_Home>/server/RoambiES/deploy`.

Make sure that the directory continues to have the `*.war` extension because the name is used by JBoss to determine how it is deployed. Without a `*.war` extension, JBoss assumes it is an EAR file, and the application will not start.

### **Creating a Windows Service**

For Windows, a service can be created for the new JBoss server instance by running the following code on the command line:

```
<JBoss_Home>/bin/RoambiES.bat install
```

### **Starting and Stopping the Roambi ES JBoss Instance**

The Roambi ES JBoss instance is now configured and can be started. For Windows, it can be started as a Windows service or by using this script:

```
<JBoss_Home>/bin/RoambiES.bat start
```

For Linux, it can be started by using this script:

```
<JBoss_Home>/bin/RoambiES.sh start
```

The Roambi Designer application is now accessible on `http://<hostname>:<port>/roambi/designer`.

The default port is **8080**.

## Chapter 4

# Deploy Roambi ES for SAS to the WebLogic Server

Deploy Roambi ES to the WebLogic Server Version 10.3.0 . . . . .	25
--	----

## Deploy Roambi ES to the WebLogic Server Version 10.3.0

Before deploying Roambi ES, the WebLogic Server must be installed and running as a service or it must be started using the WebLogic Server Administration Console.

To deploy the Roambi WAR file to the WebLogic Server:

1. Log in to the WebLogic Server Administration Console (<http://<hostname>:7501/console>).
2. In the Domain Structure panel, under **SASDomain**, select **Deployments**.
3. In the Change Center panel, click **Lock & Edit**.
4. Click **Install**.
5. Browse to find the location of the **roambi.war** folder.  
*Note:* This must be an uncompressed WAR file.
6. Click **Next**.
7. Select the **Install this deployment as an application** radio button and click **Next**.
8. In the Install Application Assistant, select the SAS server (for example, **SASServer1**), and click **Next**.
9. Select the **I will make the deployment accessible from the following location** radio button and click **Next**.
10. Click **Finish** and then click **Save**.
11. In the Change Center panel, click **Activate Changes**.
12. In the Domain Structure panel, under **SASDomain**, select **Deployments**.
13. Select the Roambi ES application. Open the **Start** menu and select **Servicing all requests**. If you are prompted to confirm the change, click **Yes**. The Roambi ES application changes to the Active state.
14. Test the application by accessing <http://<server>:7001/roambi/designer/>.



# Index

---

## A

application server connector [13](#)

## B

BlackBerry mobile devices  
prerequisites [3](#)

## C

configuring  
application server connector [13](#)  
Roambi ES connection [9](#)  
SMTP server [10](#)

## H

hardware requirements [2](#)

## I

installing Roambi ES for SAS  
on Linux [7](#)  
on Windows [5](#)

## J

Java  
heap size [10](#)  
memory parameters [10](#)  
permanent generation size [11](#)  
JBoss 4.x [11](#)  
on Linux [13](#)  
on Windows [13](#)

## M

mobile device requirements [2](#)

## O

operating environment requirements [2](#)

## R

requirements  
hardware [2](#)  
mobile devices [2](#)  
operating environment [2](#)  
SAS software [2](#)  
Roambi ES  
deploying to SAS middle tier host [15](#)  
deploying to separate host [18](#)  
extracting [8](#)  
license file [10](#)  
Setup Wizard [5](#)  
WebLogic server [25](#)

## S

SAS  
middle tier host [15](#)  
required version [2](#)  
SMTP server, configuring [10](#)

## T

Tomcat  
special configuration information [14](#)  
versions 5.x and 6.x on Linux [11](#), [13](#)  
versions 5.x and 6.x on Windows [11](#),  
[13](#)

## W

WebLogic  
deploying Roambi ES [25](#)  
version 10.3.x on Linux [12](#), [14](#)  
version 10.3.x on Windows [11](#), [14](#)  
Windows service, creating [24](#)

