# Contents

*About This Document* ................................................................. v

**Chapter 1 • Overview** ........................................................... 1  
What Is Roambi ES for SAS? ...................................................... 1  
Prerequisites and Supporting Software ........................................ 2  
Required Hardware and Operating Environments ............................ 2  
Migrating to a New Release ..................................................... 2

**Chapter 2 • Deploy Roambi ES for SAS** ..................................... 3  
Installing and Configuring Roambi ES for SAS ............................... 3  
Configuring the Application Server ............................................. 3

**Chapter 3 • Deploy Roambi ES for SAS to the Tomcat Application Server** ............ 5  
Deploying to the Apache Tomcat 6.0.32 or 7.0.x Application Server .......... 5

**Chapter 4 • Deploy Roambi ES for SAS to the JBoss Application Server** ............... 7  
Deploying to a JBoss 4.2.3 Community Edition Server ........................ 7

**Chapter 5 • Deploy Roambi ES for SAS to the Oracle WebLogic Server** ................. 15  
Deploying to the Oracle WebLogic Server Version 10.3.5 ...................... 15
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.
What Is Roambi ES for SAS?

Roambi ES for SAS integrates Roambi ES4 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 information about configuring the Roambi ES for SAS deployment. It addresses deploying Roambi ES for SAS with an Apache Tomcat Application Server, a JBoss Application Server, or an Oracle WebLogic Server.


The Roambi ES4 Administration Guide for SAS that is provided by MeLLmo and is installed with Roambi ES contains the following information:

- configuring and customizing the appearance of Roambi
- managing the Roambi portal
- configuring and managing user accounts and security
- improving performance by balancing loads, configuring firewall placement, and configuring SSL
- running Roambi batch processes
- providing troubleshooting information

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 ES4 Installation Guide for SAS* 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 9.3 Enterprise BI Server through a web service extension to SAS Web Report Studio 4.31, or to a SAS 9.4 Enterprise BI Server through a web service extension to SAS Web Report Studio 4.4.

For information about prerequisites, see the *Roambi ES4 Installation Guide for SAS* that is provided by MeLLmo.

SAS Software


Required Hardware and Operating Environments

For information about hardware requirements, supported operating environments, and iOS device requirements, see the *Roambi ES4 Installation Guide for SAS* that is provided by MeLLmo.

Note: Roambi ES does not support BlackBerry devices.

Migrating to a New Release

When you upgrade to a new release of Roambi ES for SAS, you typically back up the Roambi MySQL database and then install the new Roambi application. However, sometimes additional installation steps are required. For more information, see the *Roambi ES4 Migration Guide for SAS* that is provided by MeLLmo.
Chapter 2
Deploy Roambi ES for SAS

Installing and Configuring Roambi ES for SAS

Roambi ES can be installed on either Windows or Linux. For step-by-step installation and configuration instructions, see the Roambi ES4 Installation Guide for SAS.

Note: If you have used a previous version of Roambi ES for SAS, see the Roambi ES4 Migration Guide for SAS that is provided by MeLLmo.

Configuring the Application Server

Roambi ES requires an application server to be installed. Roambi ES can use an Apache Tomcat, a JBoss, or an Oracle WebLogic application server. For information about the required versions of the application server, see the Roambi ES4 Installation Guide for SAS that is provided by MeLLmo.

Configuration of the individual application servers varies.

• For information about the Apache Tomcat Application server, see “Deploying to the Apache Tomcat 6.0.32 or 7.0.x Application Server” on page 5.

• For information about the JBoss Application Server, see “Deploying to a JBoss 4.2.3 Community Edition Server” on page 7.

• For information about the Oracle WebLogic Server, see “Deploying to the Oracle WebLogic Server Version 10.3.5” on page 15.
Chapter 3
Deploy Roambi ES for SAS to the Tomcat Application Server

Deploying to the Apache Tomcat 6.0.32 or 7.0.x Application Server

After installing Roambi ES for SAS, you need to modify your Apache Tomcat application server to use Roambi. For step-by-step instructions, see “Modifying Apache Tomcat for Roambi” topic in Roambi ES4 Installation Guide for SAS that is provided by MeLLmo and installed with Roambi ES.
Chapter 4
Deploy Roambi ES for SAS to the JBoss Application Server

Deploying to a JBoss 4.2.3 Community Edition Server

Prepare the Roambi ES JBoss Server Instance

The JBoss instance needs to be configured from the initial installation of JBoss.

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:

   ```xml
   <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.XMLOutputStoreFactory
     </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:

   ```xml
   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
   
   ```bash
   set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m
   ```
   
   with
   
   ```bash
   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>>`).

```bash
@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
```
net stop "%SERVICENAME%"
if %ERRORLEVEL% == 0 echo JBoss server stopped as a service
goto end
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
net pause "%SERVICENAME%"
if %ERRORLEVEL% == 0 echo JBoss server paused as a service
goto end
net continue "%SERVICENAME%"
if %ERRORLEVEL% == 0 echo JBoss server resumed as a service
goto end
sc interrogate "%SERVICENAME%"
goto end

echo USAGE: %0 [{install ^| start ^| stop ^| restart ^| pause ^| resume ^| remove ^| status}]
goto end

set JAVA_HOME=
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 %*

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

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 ES4 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
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
wrapper.java.classpath.1=<<<JBOSS_HOME>>>/service/wrapper.jar

# 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=<<JBOS_HOME>>/bin/run.jar

# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path.1=<<JBOS_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
#**************************************************************************************
For a Linux deployment, create this script to start and stop the Roambi ES JBoss instance:

```
#!/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 is running (pid $pid)"
            else
                echo "JBoss is not running"
            fi
        else
            echo "JBoss is not running"
        fi
    ;;
    *)
        echo "Invalid argument: $arg"
        exit 1
    *)
    esac
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:

```bash
<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:

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

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

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

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

```bash
exit 0
```
The default port is 8080.
Deploy Roambi ES for SAS to the Oracle WebLogic Server

Deploying to the Oracle WebLogic Server Version 10.3.5

Roambi ES should not be installed on the same Oracle WebLogic server instance as any SAS web application.

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:

2. In the Domain Structure panel, under Domain, 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. In the Install Application Assistant, select the server that Roambi will be deployed to.
10. Click Finish and then click Save.
11. In the Change Center panel, click Activate Changes.
12. In the Domain Structure panel, under Domain, 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.