



Installation Guide

OpenAM 10.1

Mark Craig
Vanessa Richie
Mike Jang

ForgeRock AS
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Copyright © 2011-2017 ForgeRock AS.

Abstract

Guide showing you how to install OpenAM. OpenAM provides open source Authentication, Authorization, Entitlement and Federation software.



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License.

To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

ForgeRock® and ForgeRock Identity Platform™ are trademarks of ForgeRock Inc. or its subsidiaries in the U.S. and in other countries. Trademarks are the property of their respective owners.

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

DejaVu Fonts

Bitstream Vera Fonts Copyright

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free . fr.

FontAwesome Copyright

Copyright (c) 2017 by Dave Gandy, <http://fontawesome.io>.

This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>.

Table of Contents

Preface	iv
1. Who Should Use this Guide	iv
2. Formatting Conventions	iv
3. Accessing Documentation Online	v
4. Using the ForgeRock.org Site	v
1. Preparing For Installation	1
1.1. Preparing a Fully-Qualified Domain Name	1
1.2. Preparing Prerequisite Software	1
1.3. Obtaining OpenAM Software	3
1.4. Preparing Apache Tomcat	4
1.5. Preparing GlassFish v2	4
1.6. Preparing OpenAM & JBoss 4 or 5	5
1.7. Preparing OpenAM & JBoss 7	6
1.8. Preparing Jetty 7	10
1.9. Preparing Oracle WebLogic	10
1.10. Preparing IBM® WebSphere®	10
2. Installing OpenAM Core Services	12
3. Installing OpenAM Tools	29
4. Installing OpenAM Distributed Authentication	33
5. Installing OpenAM Client SDK Samples	36
6. Customizing the OpenAM End User Pages	39
6.1. How OpenAM Looks Up UI Files	41
7. Setting Up OpenAM Session Failover	44
8. Removing OpenAM Software	46
Index	48

Preface

This guide shows you how to install core OpenAM services for access and federation management. Unless you are planning a throwaway evaluation or test installation, read the *Release Notes* before you get started.

1. Who Should Use this Guide

This guide is written for anyone installing OpenAM to manage and to federate access to web applications and web based resources.

This guide covers the install, upgrade, and removal (a.k.a. uninstall) procedures that you theoretically perform only once per version. This guide aims to provide you with at least some idea of what happens behind the scenes when you perform the steps.

You do not need to be an OpenAM wizard to learn something from this guide, though a background in access management and maintaining web application software can help. You do need some background in managing services on your operating systems and in your application servers. You can nevertheless get started with this guide, and then learn more as you go along.

2. Formatting Conventions

Most examples in the documentation are created in GNU/Linux or Mac OS X operating environments. If distinctions are necessary between operating environments, examples are labeled with the operating environment name in parentheses. To avoid repetition file system directory names are often given only in UNIX format as in `/path/to/server`, even if the text applies to `C:\path\to\server` as well.

Absolute path names usually begin with the placeholder `/path/to/`. This path might translate to `/opt/`, `C:\Program Files\`, or somewhere else on your system.

Command-line, terminal sessions are formatted as follows:

```
$ echo $JAVA_HOME
/path/to/jdk
```

Command output is sometimes formatted for narrower, more readable output even though formatting parameters are not shown in the command.

Program listings are formatted as follows:

```
class Test {  
    public static void main(String [] args) {  
        System.out.println("This is a program listing.");  
    }  
}
```

3. Accessing Documentation Online

ForgeRock publishes comprehensive documentation online:

- The [ForgeRock Knowledge Base](#) offers a large and increasing number of up-to-date, practical articles that help you deploy and manage ForgeRock software.

While many articles are visible to community members, ForgeRock customers have access to much more, including advanced information for customers using ForgeRock software in a mission-critical capacity.

- ForgeRock product documentation, such as this document, aims to be technically accurate and complete with respect to the software documented. It is visible to everyone and covers all product features and examples of how to use them.

4. Using the ForgeRock.org Site

The [ForgeRock.org](#) site has links to source code for ForgeRock open source software, as well as links to the ForgeRock forums and technical blogs.

If you are a *ForgeRock customer*, raise a support ticket instead of using the forums. ForgeRock support professionals will get in touch to help you.

Chapter 1

Preparing For Installation

This chapter covers prerequisites for installing OpenAM software, including how to prepare your application server to run OpenAM, and how to prepare your identity repository to handle OpenAM identities.

1.1. Preparing a Fully-Qualified Domain Name

OpenAM requires that you provide the fully-qualified domain name (FQDN) when you configure it. Before you set up OpenAM, be sure that your system has an FQDN such as `openam.example.com`. For evaluation purposes, you can give your system an alias using the `/etc/hosts` file on UNIX systems or `%SystemRoot%\system32\drivers\etc\hosts` on Windows. For deployment, make sure the FQDN is properly assigned for example using DNS.

1.2. Preparing Prerequisite Software

OpenAM core services require that you install a Java runtime environment and an application container before you begin OpenAM installation.

- A Java 6 runtime environment

Check the output of `java -version` to make sure your the version is supported according to the *Release Notes* section on *Java Requirements* in the *Release Notes*.

- A supported application server as the deployment container

See the *Release Notes* in the *Release Notes* for a list.

If a Java Security Manager is enabled for your deployment container, add permissions before installing OpenAM.

- A supported identity repository for user identity data

For evaluation you can use the embedded OpenDJ LDAP directory server, which ships as part of OpenAM and that you can install as part of the OpenAM configuration process. Evaluation does not therefore require a separate user data or configuration store.

When moving to put OpenAM in production use a separate user data store in most cases. Also if you have more than four OpenAM servers in your production deployment, then use a separate LDAP

directory server for the configuration store as well. See the *Release Notes* in the *Release Notes* for a list of supported external stores.

Examples in the documentation that show a separate user data repository use OpenDJ unless otherwise indicated. In such examples OpenDJ is installed on `openam.example.com`, and listens on the following ports.

- Port 1389 for LDAP requests and StartTLS
- Port 1636 for LDAP requests over SSL
- Port 4444 for administrative traffic

The script `/etc/init.d/opendj`, created with the OpenDJ **create-rc-script** command, manages the service at system startup and shutdown. This script assumes you run OpenDJ as the user `opendj`.

```
#!/bin/sh
#...
# chkconfig: 345 95 5
# description: Control the OpenDJ Directory Server

# Set the path to the OpenDJ instance to manage
INSTALL_ROOT="/path/to/OpenDJ"
export INSTALL_ROOT

cd ${INSTALL_ROOT}

# Determine what action should be performed on the server
case "${1}" in
start)
    /bin/su opendj -c "${INSTALL_ROOT}/bin/start-ds --quiet"
    exit $?
    ;;
stop)
    /bin/su opendj -c "${INSTALL_ROOT}/bin/stop-ds --quiet"
    exit $?
    ;;
restart)
    /bin/su opendj -c "${INSTALL_ROOT}/bin/stop-ds --restart --quiet"
    exit $?
    ;;
*)
    echo "Usage: $0 { start | stop | restart }"
    exit 1
    ;;
esac
```

The Example.com sample data loaded into OpenDJ are taken from the file, `Example.ldif`, provided with the server.

See the *OpenDJ Installation Guide* for detailed installation instructions.

1.3. Obtaining OpenAM Software

Download OpenAM and policy agent releases from <https://backstage.forgerock.com/downloads/>.

For this release of OpenAM core services, you can download the entire package as a .zip archive, only the OpenAM .war file, only administrative tools, or only the configurator tool.

After you download the .zip archive, create a new `openam` folder, and unzip the archive to access the content:

```
$ cd ~/Downloads
$ mkdir openam ; cd openam
$ unzip ~/Downloads/openam_10.1.0.zip.zip
```

When you unzip the archive of the entire package, you get `ldif`, `license`, and `legal` directories in addition to the following files. See the *File Layout* reference in the *Reference* for details.

`openam-clientsdk-10.1.0-Xpress.jar`

The OpenAM Java client SDK library

`openam-distauth-10.1.0-Xpress.war`

The deployable .war file for distributed authentication

`openam-distribution-diagnostics-10.1.0-Xpress.zip`

The .zip file with the diagnostic tools to help troubleshoot deployment issue.

`openam-distribution-fedlet-unconfigured-10.1.0-Xpress.zip`

The .zip that contains the lightweight service provider implementations that you can embed in your Java EE or ASP.NET applications to enable it to use federated access management

`openam-distribution-ssoadmintools-10.1.0-Xpress.zip`

The .zip file that contains tools to manage OpenAM from the command line

`openam-distribution-ssoconfiguratortools-10.1.0-Xpress.zip`

The .zip file that contains tools to configure OpenAM from the command line

`openam-server-10.1.0-Xpress.war`

The deployable .war file

`openam-server-only-10.1.0-Xpress.war`

The deployable .war file when you want to deploy OpenAM server without the OpenAM console

1.4. Preparing Apache Tomcat

OpenAM examples often use Apache Tomcat as the deployment container. Tomcat is installed on `openam.example.com`, and listens on the default ports, with no Java Security Manager enabled. The script `/etc/init.d/tomcat` manages the service at system startup and shutdown. This script assumes you run OpenAM as the user `openam`.

OpenAM core services require a minimum JVM heap size of 1 GB, and a permanent generation size of 256 MB.

```
#!/bin/sh
#
# tomcat
#
# chkconfig: 345 95 5
# description: Manage Tomcat web application container
CATALINA_HOME="/path/to/tomcat"
export CATALINA_HOME
JAVA_HOME=/path/to/jdk1.6
export JAVA_HOME
JAVA_OPTS="-Xmx1024m -XX:MaxPermSize=256m"
export JAVA_OPTS

case "${1}" in
start)
    /bin/su openam -c "${CATALINA_HOME}/bin/startup.sh"
    exit ${?}
    ;;
stop)
    /bin/su openam -c "${CATALINA_HOME}/bin/shutdown.sh"
    exit ${?}
    ;;
*)
    echo "Usage: $0 { start | stop }"
    exit 1
    ;;
esac
```

1.5. Preparing GlassFish v2

Before you deploy OpenAM, update these JVM options. These settings are accessible in the administration console under Application Server > JVM Settings > JVM Options.

-server

Use `-server` rather than `-client`.

-XX:MaxPermSize=256m

Set the permanent generation size to 256 MB.

-Xmx1024m

Make sure OpenAM has at least 1 GB heap.

1.6. Preparing OpenAM & JBoss 4 or 5

OpenAM must be able to store its configuration between restarts. If you plan to deploy OpenAM as a single archive file, then unpack the .war, edit `WEB-INF/classes/bootstrap.properties` to set the `configuration.dir` property to the location where OpenAM has write access to store its configuration.

```
$ mkdir openam
$ cd openam
$ jar -xf ~/Downloads/openam/openam-server-10.1.0-Xpress.war
$ vi WEB-INF/classes/bootstrap.properties
$ grep ^config WEB-INF/classes/bootstrap.properties
configuration.dir=/home/jboss-user/openam
```

Also, OpenAM .jar libraries that conflict with JBoss libraries must be loaded first. Add a `WEB-INF/jboss-web.xml` to ensure this happens. (If you deploy the exploded .war, you also need to add this file.)

```
$ vi WEB-INF/jboss-web.xml
$ cat WEB-INF/jboss-web.xml
```

```
<!DOCTYPE jboss-web PUBLIC
"-//JBoss//DTD Web Application 5.0//EN"
"http://www.jboss.org/j2ee/dtd/jboss-web_5_0.dtd">
<jboss-web>
  <class-loading java2ClassLoadingCompliance='true'>
    <loader-repository>
      jbia.loader:loader=opensso
      <loader-repository-config>java2ParentDelegaton=true</loader-repository-config>
    </loader-repository>
  </class-loading>
</jboss-web>
```

Repack the .war file that you can then deploy.

```
$ jar -cf ../openam.war *
```

Before you deploy OpenAM, update these JVM options.

-server

Use `-server` rather than `-client`.

-XX:MaxPermSize=256m

Set the permanent generation size to 256 MB.

-Xmx1024m

Make sure OpenAM has at least 1 GB heap.

1.7. Preparing OpenAM & JBoss 7

OpenAM must be able to store its configuration between restarts. If you plan to deploy OpenAM as a single archive file, then unpack the .war, edit `WEB-INF/classes/bootstrap.properties` to set the `configuration.dir` property to the location where OpenAM has write access to store its configuration, and then repack the .war.

```
$ mkdir openam
$ cd openam
$ jar -xf ~/Downloads/openam/openam-server-10.1.0-Xpress.war
$ vi WEB-INF/classes/bootstrap.properties
$ grep ^config WEB-INF/classes/bootstrap.properties
configuration.dir=/home/jboss-user/openam
$ jar -cf ../openam.war *
```

Procedure 1.1. Preparing JBoss 7

1. Stop JBoss.
2. Add the Sun x509 security module path to the JBoss 7 configuration.

The following example uses JBoss 7.1.1.

```
$ cp /path/to/jboss7/modules/sun/jdk/main/module.xml
/path/to/jboss7/modules/sun/jdk/main/module.orig
$ vi /path/to/jboss7/modules/sun/jdk/main/module.xml
$ diff -c /path/to/jboss7/modules/sun/jdk/main/module.orig
/path/to/jboss7/modules/sun/jdk/main/module.xml
*** /path/to/jboss7/modules/sun/jdk/main/module.orig    ...
--- /path/to/jboss7/modules/sun/jdk/main/module.xml    ...
*****
*** 38,43 ****
--- 38,44 ----
<path name="com/sun/security/auth"/>
<path name="com/sun/security/auth/login"/>
<path name="com/sun/security/auth/module"/>
+       <path name="sun/security/x509"/>
<path name="sun/misc"/>
<path name="sun/io"/>
<path name="sun/nio"/>
```

3. Disable modules that conflict with OpenAM.
 - The following example uses JBoss 7.1.1 standalone.

```
$ cp /path/to/jboss7/standalone/configuration/standalone.xml
/path/to/jboss7/standalone/configuration/standalone.orig
$ vi /path/to/jboss7/standalone/configuration/standalone.xml
$ diff -c /path/to/jboss7/standalone/configuration/standalone.orig
/path/to/jboss7/standalone/configuration/standalone.xml
*** /path/to/jboss7/standalone/configuration/standalone.orig    ...
--- /path/to/jboss7/standalone/configuration/standalone.xml    ...
*****
*** 9,15 ****
<extension module="org.jboss.as.deployment-scanner"/>
```

```

<extension module="org.jboss.as.ee"/>
<extension module="org.jboss.as.ejb3"/>
-   <extension module="org.jboss.as.jaxrs"/>
<extension module="org.jboss.as.jdr"/>
<extension module="org.jboss.as.jmx"/>
<extension module="org.jboss.as.jpas"/>
--- 9,14 ----
*****
*** 24,30 ***
<extension module="org.jboss.as.threads"/>
<extension module="org.jboss.as.transactions"/>
<extension module="org.jboss.as.web"/>
-   <extension module="org.jboss.as.webservices"/>
<extension module="org.jboss.as.weld"/>
</extensions>

--- 23,28 ----
*****
*** 163,169 ***
</local-cache>
</cache-container>
</subsystem>
-   <subsystem xmlns="urn:jboss:domain:jaxrs:1.0"/>
<subsystem xmlns="urn:jboss:domain:jca:1.1">
<archive-validation enabled="true" fail-on-error="true"
    fail-on-warn="false"/>
<bean-validation enabled="true"/>
--- 161,166 ----
*****
*** 262,277 ***
<alias name="example.com"/>
</virtual-server>
</subsystem>
-   <subsystem xmlns="urn:jboss:domain:webservices:1.1">
-       <modify-wsdl-address>true</modify-wsdl-address>
-       <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
-       <endpoint-config name="Standard-Endpoint-Config"/>
-       <endpoint-config name="Recording-Endpoint-Config">
-           <pre-handler-chain name="recording-handlers"
-               protocol-bindings="##SOAP11_HTTP ##SOAP11_HTTP_MTOM
-               ##SOAP12_HTTP ##SOAP12_HTTP_MTOM">
-               <handler name="RecordingHandler"/>
-           </pre-handler-chain>
-       </endpoint-config>
-   </subsystem>
<subsystem xmlns="urn:jboss:domain:weld:1.0"/>
</profile>

--- 259,264 ----

```

- The following example uses JBoss 7.1.1 for a managed domain.

```

$ cp /path/to/jboss7/domain/configuration/domain.xml
  /path/to/jboss7/domain/configuration/domain.orig
$ vi /path/to/jboss7/domain/configuration/domain.xml
$ diff -c /path/to/jboss7/domain/configuration/domain.orig
  /path/to/jboss7/domain/configuration/domain.xml
*** /path/to/jboss7/domain/configuration/domain.orig    ...

```

```

--- /path/to/jboss7/domain/configuration/domain.xml    ...
*****
*** 11,17 ****
<extension module="org.jboss.as.ejb3"/>
<extension module="org.jboss.as.jacorb"/>
<extension module="org.jboss.as.jaxr"/>
-     <extension module="org.jboss.as.jaxrs"/>
<extension module="org.jboss.as.jdr"/>
<extension module="org.jboss.as.jmx"/>
<extension module="org.jboss.as.jpas"/>
--- 11,16 ----
*****
*** 29,35 ****
<extension module="org.jboss.as.threads"/>
<extension module="org.jboss.as.transactions"/>
<extension module="org.jboss.as.web"/>
-     <extension module="org.jboss.as.webservices"/>
<extension module="org.jboss.as.weld"/>
</extensions>
<system-properties>
--- 28,33 ----
*****
*** 146,152 ****
</local-cache>
</cache-container>
</subsystem>
-     <subsystem xmlns="urn:jboss:domain:jaxrs:1.0"/>
<subsystem xmlns="urn:jboss:domain:jca:1.1">
<archive-validation enabled="true" fail-on-error="true"
    fail-on-warn="false"/>
<bean-validation enabled="true"/>
--- 144,149 ----
*****
*** 246,261 ****
<alias name="example.com"/>
</virtual-server>
</subsystem>
-     <subsystem xmlns="urn:jboss:domain:webservices:1.1">
-         <modify-wsdl-address>true</modify-wsdl-address>
-         <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
-         <endpoint-config name="Standard-Endpoint-Config"/>
-         <endpoint-config name="Recording-Endpoint-Config">
-             <pre-handler-chain name="recording-handlers"
-                 protocol-bindings="##SOAP11_HTTP ##SOAP11_HTTP_MTOM
-                 ##SOAP12_HTTP ##SOAP12_HTTP_MTOM">
-                 <handler name="RecordingHandler"/>
-             </pre-handler-chain>
-         </endpoint-config>
-     </subsystem>
<subsystem xmlns="urn:jboss:domain:weld:1.0"/>
</profile>
<profile name="ha">
--- 243,248 ----
*****
*** 544,559 ****
<alias name="example.com"/>
</virtual-server>
</subsystem>
-     <subsystem xmlns="urn:jboss:domain:webservices:1.1">

```

```

-         <modify-wsdl-address>true</modify-wsdl-address>
-         <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
-         <endpoint-config name="Standard-Endpoint-Config"/>
-         <endpoint-config name="Recording-Endpoint-Config">
-             <pre-handler-chain name="recording-handlers"
-                 protocol-bindings="##SOAP11_HTTP ##SOAP11_HTTP_MTOM
-                                     ##SOAP12_HTTP ##SOAP12_HTTP_MTOM">
-                 <handler name="RecordingHandler"/>
-             </pre-handler-chain>
-         </endpoint-config>
-     </subsystem>
<subsystem xmlns="urn:jboss:domain:weld:1.0"/>
</profile>
<profile name="full">
--- 531,536 ----
*****
*** 859,874 ****
<alias name="example.com"/>
</virtual-server>
</subsystem>
-     <subsystem xmlns="urn:jboss:domain:webservices:1.1">
-         <modify-wsdl-address>true</modify-wsdl-address>
-         <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
-         <endpoint-config name="Standard-Endpoint-Config"/>
-         <endpoint-config name="Recording-Endpoint-Config">
-             <pre-handler-chain name="recording-handlers"
-                 protocol-bindings="##SOAP11_HTTP ##SOAP11_HTTP_MTOM
-                                     ##SOAP12_HTTP ##SOAP12_HTTP_MTOM">
-                 <handler name="RecordingHandler"/>
-             </pre-handler-chain>
-         </endpoint-config>
-     </subsystem>
<subsystem xmlns="urn:jboss:domain:weld:1.0"/>
</profile>
<profile name="full-ha">
--- 836,841 ----
*****
*** 1275,1290 ****
<alias name="example.com"/>
</virtual-server>
</subsystem>
-     <subsystem xmlns="urn:jboss:domain:webservices:1.1">
-         <modify-wsdl-address>true</modify-wsdl-address>
-         <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
-         <endpoint-config name="Standard-Endpoint-Config"/>
-         <endpoint-config name="Recording-Endpoint-Config">
-             <pre-handler-chain name="recording-handlers"
-                 protocol-bindings="##SOAP11_HTTP ##SOAP11_HTTP_MTOM
-                                     ##SOAP12_HTTP ##SOAP12_HTTP_MTOM">
-                 <handler name="RecordingHandler"/>
-             </pre-handler-chain>
-         </endpoint-config>
-     </subsystem>
<subsystem xmlns="urn:jboss:domain:weld:1.0"/>
</profile>
</profiles>
--- 1242,1247 ----

```

4. Before you deploy OpenAM, update these JVM options.

`-Dorg.apache.tomcat.util.http.ServerCookie.ALWAYS_ADD_EXPIRES=true`

Make sure that headers include the `Expires` attribute rather than only `Max-Age`, as some versions of Internet Explorer do not support `Max-Age`.

`-server`

Use `-server` rather than `-client`.

`-XX:MaxPermSize=256m`

Set the permanent generation size to 256 MB.

`-Xmx1024m`

Make sure OpenAM has at least 1 GB heap.

1.8. Preparing Jetty 7

When you deploy OpenAM, make sure you start Jetty with enough memory. OpenAM core services require a minimum JVM heap size of 1 GB, and a permanent generation size of 256 MB.

```
$ cd /path/to/jetty
$ java -server -Xmx1024m -XX:MaxPermSize=256m -jar start.jar
```

1.9. Preparing Oracle WebLogic

Before you deploy OpenAM, update these JVM options.

`-server`

Use `-server` rather than `-client`.

`-XX:MaxPermSize=256m`

Set the permanent generation size to 256 MB.

`-Xmx1024m`

Make sure OpenAM has at least 1 GB heap.

1.10. Preparing IBM[®] WebSphere[®]

Before you deploy OpenAM, add these JVM parameters in the Administrator console.

Generic JVM arguments

-DamCryptoDescriptor.provider=IBMJCE

-DamKeyGenDescriptor.provider=IBMJCE

Make sure OpenAM has at least 1 GB heap.

Chapter 2

Installing OpenAM Core Services

This chapter covers tasks required for a full install of OpenAM core services including the OpenAM console and installation of only the core services.

To manage access to resources on other servers, you can use either OpenIG or one of the OpenAM policy agents. OpenIG is a high-performance reverse proxy server with specialized session management and credential replay functionality. It can function as a standards-based policy enforcement point. Policy agents provide policy enforcement on supported web servers and Java EE containers, and are tightly integrated with OpenAM. See the *Policy Agent Installation Guide* for instructions on installing OpenAM agents in supported web servers and Java EE application containers.

The `openam.war` file contains all OpenAM server components and samples. How you deploy the `.war` file depends on your web application container.

Table 2.1. Deciding How To Install OpenAM

If you want to...	Then see...
Install quickly for evaluation using default settings	Procedure 2.1, "To Deploy OpenAM" and Procedure 2.2, "To Configure OpenAM With Defaults (For Evaluation)"
Install core OpenAM and the console, choosing settings	Procedure 2.1, "To Deploy OpenAM" and Procedure 2.4, "To Configure OpenAM"
Install additional instance of OpenAM	Procedure 2.1, "To Deploy OpenAM", Procedure 2.4, "To Configure OpenAM", and Procedure 2.5, "Configuring an Additional Instance"
Erase the configuration and start over	Procedure 2.3, "To Delete an OpenAM Configuration Before Redeploying"
Perform a command-line install	Procedure 2.1, "To Deploy OpenAM" and <i>To Set Up Configuration Tools</i>
Set up high availability in a site configuration	Procedure 2.1, "To Deploy OpenAM" with either Procedure 2.4, "To Configure OpenAM" or Procedure 2.5, "Configuring an Additional Instance"
Install OpenAM with no console	Table 2.2, "Determine Which War File to Deploy", then Procedure 2.1, "To Deploy OpenAM"
Install <code>ssoadm</code> and other tools	<i>Installing OpenAM Tools</i> , or OpenAM <code>ssoadm.jsp</code> in the <i>Administration Guide</i> in the <i>Administration Guide</i>

If you want to...	Then see...
Install OpenAM in your DMZ	<i>Installing OpenAM Distributed Authentication</i>
Skin OpenAM for your organization	<i>Customizing the OpenAM End User Pages</i>
Uninstall OpenAM	<i>Removing OpenAM Software</i>

Select the `.war` file based on the type of deployment you need, as defined in the following table.

Table 2.2. Determine Which War File to Deploy

If you want to...	Use...
Install core OpenAM and the console	<code>openam-server-10.1.0-Xpress.war</code>
Install OpenAM with distributed authentication	<code>openam-distauth-10.1.0-Xpress.war</code>
Install OpenAM with no console	<code>openam-server-only-10.1.0-Xpress.war</code>

Procedure 2.1. To Deploy OpenAM

The `openam-server-10.1.0-Xpress.war` file contains all OpenAM server components and samples. How you deploy the `.war` file depends on your web application container.

1. Deploy the `.war` file on your container.

For example, copy the file to deploy on Apache Tomcat.

```
$ cp openam/openam-server-10.1.0-Xpress.war
/path/to/tomcat/webapps/openam.war
```

Note

By adding `openam.war` to the end of the path, the name of the `.war` file will be changed from `openam-server-only-10.1.0-Xpress.war` to `openam.war` to make deployment easier.

2. Check that you see the initial configuration screen in your browser at `openam.example.com:8080/openam`.

Note

You should NOT use localhost domain for accessing OpenAM, not even for testing purposes, because OpenAM strongly relies on browser cookies. Also the chosen domain name should contain at least 2 '.' (dot) characters, like `openam.example.com`. This gives your instance a domain, `example`, and a subdomain, `openam`. Sub-domains provide an extra layer of organization for different department or types of users or clients.

It also provides a level of control over your top-level domain that you would not have if you used .com or .org.



Copyright © 2010-2012 ForgeRock AS, Philip Pedersens vei 1, 1366 Lysaker, Norway. All rights reserved. Licensed for use under the Common Development and Distribution License (CDDL), see <http://www.forgerock.com/license/CDDLv1.0.html> for details. This software is based on the OpenSSO/OpenAM open source project and the source includes the copyright works of other authors, granted for use under the CDDL. This distribution may include other materials developed by third parties. All Copyrights and Trademarks are property of their owners.

Configuration Options

Please select a configuration option.

Default Configuration

Enter only the passwords for the default administrator and the agent accessor. All other data is configured using default parameters. This option should be used primarily for evaluation or development purposes.
[Create Default Configuration](#)

Custom Configuration

Allows you to specify all configuration parameters including the type of data store, encryption properties, user data store, etc. This option has the most flexibility in setting up your installation.
[Create New Configuration](#)

Procedure 2.2. To Configure OpenAM With Defaults (For Evaluation)

The default configuration option will basically configure the embedded OpenDJ instance on default ports (if the ports are already in use, OpenAM will look for a free port) as both configuration store and identity store. The install will create a standalone OpenAM instance using the subset of the fully qualified hostname as the cookie domain. In the example `.example.com` the cookie domain is set to `.example.com`.

Note

The configuration settings are saved to the `$HOME` of the user running the web application container. If you would like the configuration files stored somewhere else, such as `/opt/openam`, you will need to run the Custom Configuration.

1. In the initial configuration screen, click [Create Default Configuration](#) under Default Configuration.
2. Provide different passwords for the default OpenAM administrator, `amadmin`, and default Policy Agent users.

The screenshot shows a dialog box titled "OpenAM Configurator" with a sub-header "Default Configuration Option". The main text reads: "Use this option for a quick setup. Only the super user name and agent user name are required. All other configuration parameters are defaulted for you. The user and agent passwords must be different values. * Indicates required field".

There are two sections for password entry:

- Default User [amAdmin]**
 - * Password: [password field] OK
 - * Confirm Password: [password field]
- Default Policy Agent [UriAccessAgent]**
 - * Password: [password field] OK
 - * Confirm Password: [password field]

At the bottom, there are two buttons: "Create Configuration" and "Cancel".

3. When the configuration completes, click Proceed to Login, and then login as OpenAM administrator with the first of the two passwords you provided.

VERSION
LOG OUT

User: amAdmin Server: ubuntu

Common Tasks
Access Control
Federation
Configuration
Sessions

Create SAMLv2 Providers

Use these work flows to create hosted or remote identity and service providers for SAMLv2 Federation.

Create Hosted Identity Provider i

Create Hosted Service Provider i

Register Remote Identity Provider i

Register Remote Service Provider i

Configure OAuth2

This task configures OAuth2 per realm. Each realm can act as an authorization server.

Configure OAuth2 i

Create Fedlet

Create a Fedlet to enable federation between an identity provider hosted on this instance of OpenAM and a remote service provider that does not have a federation solution. Before beginning, a hosted identity provider must be configured.

Create Fedlet i

Configure Google Apps

Integrate OpenAM with Google Apps web applications to create a single sign-on environment. Before beginning, a hosted identity provider and Circle of Trust must be configured.

Configure Google Apps i

Configure Salesforce CRM

Integrate OpenAM with Salesforce CRM to create a single sign-on environment. Before beginning, a SAMLv2 hosted identity provider and Circle of Trust must be configured.

Configure Salesforce CRM i

Test Federation Connectivity

Use this automated test to determine if federation connections are being made successfully, or to identify where issues might be located.

Test Federation Connectivity i

Get Product Documentation

Launch the OpenAM product documentation page.

Get Product Documentation i

Procedure 2.3. To Delete an OpenAM Configuration Before Redeploying

1. Stop the OpenAM web application to clear the configuration held in memory.

The following example shuts down Tomcat configured as described above.

```

$ /etc/init.d/tomcat stop
Password:
Using CATALINA_BASE:   /path/to/tomcat
Using CATALINA_HOME:   /path/to/tomcat
Using CATALINA_TMPDIR: /path/to/tomcat/temp
Using JRE_HOME:        /path/to/jdk1.6/jre
Using CLASSPATH:
/path/to/tomcat/bin/bootstrap.jar:/path/to/tomcat/bin/tomcat-juli.jar
    
```

2. Delete OpenAM configuration files, by default under the `$HOME` of the user running the web application container.

```
$ rm -rf $HOME/openam $HOME/.openamcfg
```

Note

When using the internal OpenAM configuration store, this step deletes the embedded directory server and all of its contents. You should always stop the application server before removing the configuration files. In case you're using external configuration store make sure you delete the entries under the configured OpenAM suffix (by default `dc=openam,dc=forgerock,dc=org`).

3. Restart the OpenAM web application.

The following example restarts the Tomcat container.

```
$ /etc/init.d/tomcat start
Password:
Using CATALINA_BASE:   /path/to/tomcat
Using CATALINA_HOME:   /path/to/tomcat
Using CATALINA_TMPDIR: /path/to/tomcat/temp
Using JRE_HOME:        /path/to/jdk1.6/jre
Using CLASSPATH:       /path/to/tomcat/bin/bootstrap.jar:/path/to/tomcat/bin/tomcat-juli.jar
```

Procedure 2.4. To Configure OpenAM

1. In the initial configuration screen, click Create New Configuration under Custom Configuration.
2. Provide a password having at least 8 characters for the OpenAM Administrator, `amadmin`.

OpenAM Configurator

Custom Configuration Option

- **General**
- 2. Server Settings
- 3. Configuration Store
- 4. User Store
- 5. Site Configuration
- 6. Agent Information
- 7. Summary

Step 1: General

Enter the password for the default user, amAdmin. The password must be at least 8 characters in length. If this configuration will be part of an existing deployment, the password you enter must match that of the original deployment.

* Indicates required field

Default User Password

Default User [amAdmin]

* Password

* Confirm Password

3. Make sure the server settings are valid for your configuration.

OpenAM Configurator [X]

Custom Configuration Option

- 1. General
- **Server Settings**
- 3. Configuration Store
- 4. User Store
- 5. Site Configuration
- 6. Agent Information
- 7. Summary

Step 2: Server Settings ⓘ

Confirm the following settings to use for the server.

* Indicates required field

Server Settings

* Server URL	<input type="text" value="http://openam.example.com:18080"/>	<input checked="" type="checkbox"/> OK
* Cookie Domain	<input type="text" value=".example.com"/>	<input checked="" type="checkbox"/> OK
* Platform Locale	<input type="text" value="en_US"/>	
* Configuration Directory	<input type="text" value="/opt/openam/instance1"/>	<input checked="" type="checkbox"/> OK

Server URL

Provide a valid URL to the base of your OpenAM web container, including a fully qualified domain name (FQDN).

In a test environment, you can fake the FQDN by adding it to your `/etc/hosts` as an alias. The following excerpt shows lines from the `/etc/hosts` file on a Linux system where OpenAM is installed.

```
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
127.0.1.1 openam openam.example.com
```

Cookie Domain

Starts with a dot (.).

Platform Locale

Supported locales include `en_US` (English), `de` (German), `es` (Spanish), `fr` (French), `ja` (Japanese), `ko` (Korean), `zh_CN` (Simplified Chinese), and `zh_TW` (Traditional Chinese).

Configuration Directory

Location on server for OpenAM configuration files. OpenAM must be able to write to this directory.

4. In the Configuration Store screen, you can accept the defaults to allow OpenAM to store configuration data in an embedded directory. The embedded directory can be configured separately to replicate data for high availability if necessary.

You can also add this OpenAM installation to an existing deployment, providing the URL to reach an existing OpenAM instance. Procedure 2.5, "Configuring an Additional Instance" provides information on setting up an additional instance of OpenAM.

Alternatively, if you already manage an OpenDJ or DSEE deployment, you can choose to store OpenAM configuration data in your existing directory service. You must, however, create the suffix to store configuration data on the directory server before you configure OpenAM. OpenAM does not create the suffix when you use an external configuration store.

Note

When you create a new OpenAM custom configuration that uses an external LDAP directory server for the configuration data store, you must use a root suffix DN with at least two domain components, such as `dc=example,dc=com`. The root suffix must be `dc`. You will not be able to move to the next configuration screen if both of these conditions are not met.

5. In the User Store screen, you configure where OpenAM looks for user identities.

OpenAM must have write access to the directory service you choose, as it adds to the directory schema needed to allow OpenAM to manage access for users in the user store.

OpenAM Configurator
✕

Custom Configuration Option

1. General
2. Server Settings
3. Configuration Store
- 4. User Store
5. Site Configuration
6. Agent Information
7. Summary

Step 4: User Data Store Settings

You can use the data store that comes with the OpenAM configuration data store, or you can use a different user data store. A good practice for setting up production environments is to use an external user data store, one that is different than the OpenAM user data store. Please note that Policy Service and LDAP Authentication Module shall be configured to use the Directory Administrator DN and Password provided here.

OpenAM User Data Store
 Other User Data Store

* Indicates required field

User Store Details

* User Data Store Type Oracle Directory Server Enterprise Edition OpenDJ
 Active Directory with Host and Port AD with Domain Name
 Active Directory Application Mode IBM Tivoli Directory Server

* SSL/TLS Enabled

* Directory Name

* Port

* Root Suffix

* Login ID

* Password OK

User Data Store Type

If you have a directory service already provisioned with users in a supported user data store, then select that type of directory from the options available.

SSL/TLS Enabled

To use a secure connection, check this box, then make sure the Port you define corresponds to the port on which the directory listens for StartTLS or SSL connections. When using this option you also need to make sure the trust store used by the JVM running OpenAM has the necessary certificates installed.

Directory Name

FQDN for the host housing the directory service

Port

LDAP directory port. The default for LDAP and LDAP with StartTLS to protect the connection is port 389. The default for LDAP over SSL is port 636. Your directory service might use a different port.

Root Suffix

Base distinguished name (DN) where user data are stored

Login ID

Directory administrator user DN. The administrator must be capable of updating schema and user data.

Password

Password for the directory administrator user

6. In the Site Configuration screen, you can set up OpenAM as part of a site where the load is balanced across multiple OpenAM servers.

For your first OpenAM installation, you can accept the defaults. If you have a load balancer, you can enable session high availability persistence. This will store sessions in case of a server failure, so that when the server is restored, users will be returned to their sessions without having to login again. If you would like to enable session high availability persistence or if you plan to setup additional instances, enter the Site Name, Load Balancer URL, and click the [Enable Session HA Persistence and Failover](#).

OpenAM Configurator
X

Custom Configuration Option

1. General
2. Server Settings
3. Configuration Store
4. User Store
- 5. Site Configuration
6. Agent Information
7. Summary

Step 5: Site Configuration

Will this instance be deployed behind a load balancer as part of a site configuration?

No
 Yes

* Indicates required field

Site Configuration Details

This is the first instance of OpenAM, and no site configurations currently exist. To create a new site configuration, provide the following information

* Site Name

* Load Balancer URL

Enable Session HA Persistence and Failover

If you plan to set up an additional instance but are not ready yet, you do not have to set it up now. The site configuration can be set up during configuration of your additional instances.

7. In the Agent Information screen, provide a password having at least 8 characters to be used by policy agents to connect to OpenAM.

OpenAM Configurator
X

Custom Configuration Option

1. General
2. Server Settings
3. Configuration Store
4. User Store
5. Site Configuration
- 6. Agent Information
7. Summary

Step 6: Default Policy Agent User

These settings are used by OpenAM policy agents for retrieving policy agent properties.

* Indicates required field

Policy Agent User

Default Policy Agent [UriAccessAgent]

* Password OK

* Confirm Password

- When the configuration completes, click Proceed to Login, and then login as OpenAM administrator.

The screenshot shows the OpenAM configuration web interface. At the top, it displays 'VERSION' and 'LOG OUT' buttons. Below that, it shows 'User: amAdmin' and 'Server: ubuntu'. The OpenAM logo is visible. A navigation bar contains tabs for 'Common Tasks', 'Access Control', 'Federation', 'Configuration', and 'Sessions'. The main content area is divided into two columns of tasks, each with a description and a button with an information icon.

Create SAMLv2 Providers
Use these work flows to create hosted or remote identity and service providers for SAMLv2 Federation.

- Create Hosted Identity Provider
- Create Hosted Service Provider
- Register Remote Identity Provider
- Register Remote Service Provider

Configure OAuth2
This task configures OAuth2 per realm. Each realm can act as an authorization server.

- Configure OAuth2

Create Fedlet
Create a Fedlet to enable federation between an identity provider hosted on this instance of OpenAM and a remote service provider that does not have a federation solution. Before beginning, a hosted identity provider must be configured.

- Create Fedlet

Configure Google Apps
Integrate OpenAM with Google Apps web applications to create a single sign-on environment. Before beginning, a hosted identity provider and Circle of Trust must be configured.

- Configure Google Apps

Configure Salesforce CRM
Integrate OpenAM with Salesforce CRM to create a single sign-on environment. Before beginning, a SAMLv2 hosted identity provider and Circle of Trust must be configured.

- Configure Salesforce CRM

Test Federation Connectivity
Use this automated test to determine if federation connections are being made successfully, or to identify where issues might be located.

- Test Federation Connectivity

Get Product Documentation
Launch the OpenAM product documentation page.

- Get Product Documentation

- Restrict permissions to the configuration directory (by default `$HOME/openam`, where `$HOME` corresponds to the user who runs the web container).

Procedure 2.5. Configuring an Additional Instance

When an additional OpenAM instance is necessary (usually for *Setting Up OpenAM Session Failover* [SFO]), it is configured separately from the first instance.

- In the initial configuration screen, click Create New Configuration under Custom Configuration.

2. Enter the same password entered during the first instance configuration for the OpenAM Administrator, `amadmin`.

Note

If you make a mistake at any time during the configuration, click on the **Previous** button and the OpenAM Configurator will return to the previous screen. When you click **Next**, the original default values will appear.

3. Make sure the server settings are valid for your configuration.

If you make changes to the default values, an x mark or a check mark will indicate if the added value is acceptable. For example, you cannot add additional instances under the same configuration directory as the first instance.

The screenshot shows the 'OpenAM Configurator' window with a title bar and a close button. Below the title bar is a header 'Custom Configuration Option'. On the left is a navigation menu with items: 1. General, 2. Server Settings (selected with a right arrow), 3. Configuration Store, 4. User Store, 5. Site Configuration, 6. Agent Information, and 7. Summary. The main content area is titled 'Step 2: Server Settings' with a help icon. Below the title is the instruction 'Confirm the following settings to use for the server.' and a legend '* Indicates required field'. A 'Server Settings' panel contains four fields: '* Server URL' with value 'http://openam.example.com:28080', '* Cookie Domain' with value 'example.com', '* Platform Locale' with value 'en_US', and '* Configuration Directory' with value '/opt/openam/instance2'.

4. Click on **Add to Existing Deployment?**. Enter the Server URL of the first instance, for example `http://openam.example.com:18080/openam`. You should be able to accept the default settings once your Server URL has been verified.

OpenAM Configurator

Custom Configuration Option

- 1. General
- 2. Server Settings
- **Configuration Store**
- 4. User Store
- 5. Site Configuration
- 6. Agent Information
- 7. Summary

Step 3: Configuration Data Store Settings

If no other OpenAM instance already exists in the environment, then choose First Instance. If one or more OpenAM instances already exist in the environment, choose Add to Existing Deployment.

First Instance Add to Existing Deployment? * Indicates required field

Configuration Store Details

* Server URL
 OK
URL of the existing OpenAM server. ex: http://server.co.com:8080/openam

New OpenAM instance port settings

* Listening Port	<input type="text" value="51389"/>	* Admin Port	<input type="text" value="5444"/>
* Replication Port	<input type="text" value="58989"/>	* JMX Port	<input type="text" value="2689"/>

The existing OpenAM instance is not set up for replication. The following ports will be used

* Admin Port	<input type="text" value="4444"/>
* Replication Port	<input type="text" value="50889"/>

5. Select **Yes** to add the load balancer with each additional instance. It is alright if you did not add the load balancer with the first instance. The *SFO* chapter describes how to add the load balancer to the first instance.

Enter the **Site Name**, for example `openamlb`, and the **Load Balancer URL**.

OpenAM Configurator
✕

Custom Configuration Option

1. General
2. Server Settings
3. Configuration Store
4. User Store
- ➔ 5. **Site Configuration**
6. Agent Information
7. Summary

Step 5: Site Configuration

Will this instance be deployed behind a load balancer as part of a site configuration?

No
 Yes

* Indicates required field

Site Configuration Details

This is the first instance of OpenAM, and no site configurations currently exist. To create a new site configuration, provide the following information

* Site Name
 OK

* Load Balancer URL
 OK

Enable Session HA Persistence and Failover OK

6. Verify the Configurator Summary Details.
7. When the configuration completes, click **Proceed to Login**, and then login as OpenAM administrator.

Now when an end user logs into a server on a load balancer, they will have a persistent session, even if the first server the user logs into fails.

Procedure 2.6. To Deploy the Core Services On Tomcat

You can deploy OpenAM core services without including the console if you install the console elsewhere, or if you plan to perform all configuration using **ssoadm** for example.

1. Deploy the `openam-server-only-10.1.0-Xpress.war` file on your container.

For example, copy the file to deploy on Apache Tomcat.

```
$ cp ~/Downloads/openam/openam-server-only-10.1.0-Xpress.war
/path/to/tomcat/webapps/coreonly.war
```


Note

By adding `coreonly.war` to the end of the path, the name of the `.war` file will be changed from `openam-server-only-10.1.0-Xpress.war` to `coreonly.war` so the deployment URI will be `/coreonly`.

2. Browse to the console application, for example `http://host.example.com:8080/coreonly/`, and configure OpenAM core services as if you were deploying with a full version.
3. Restrict permissions to the `$HOME/coreonly` configuration directory, where `$HOME` corresponds to the user who runs the web container.

Chapter 3

Installing OpenAM Tools

OpenAM tools are found in three `.zip` files inside the archive of the entire OpenAM zip download package, `openam_10.1.0.zip`. When you unpack the full zip package, you find these files.

`openam-distribution-ssoadmintools-10.1.0-Xpress.zip`

Administration tools: **ampassword**, **ssoadm** and **amverifyarchive**

This administration tools package can be downloaded separately as `ssoAdminTools_10.1.0.zip`.

`openam-distribution-ssoconfigurortools-10.1.0-Xpress.zip`

Tool to check for command line installation, upgrade, and initial configuration, as well as serving as a command line alternative to using the GUI configuration wizard

This configurator package can be downloaded separately as `ssoconfigurortools_10.1.0.zip`.

`openam-distribution-diagnostics-10.1.0-Xpress.zip`

Tool to check for problems with your deployment and configuration

Procedure 3.1. To Set Up Administration Tools

1. Make sure OpenAM is installed on the system before proceeding.
2. Make sure the `JAVA_HOME` environment variable is properly set.

```
$ echo $JAVA_HOME  
/path/to/jdk1.6
```

3. Unpack the tools from where you unzipped OpenAM.

```
$ cd /path/to/openam-tools/admin  
$ unzip ~/Downloads/openam/  
openam-distribution-ssoadmintools-10.1.0-Xpress.zip  
...  
  inflating: template/windows/bin/amverifyarchive.bat.template  
  inflating: template/windows/bin/ssoadm.bat.template
```

4. Run the **setup** utility (**setup.bat** on Windows), providing the path to the directory where OpenAM configuration files are located, and where you want debug and log information to be located.

```
$ ./setup
Path to config files of OpenAM server [/home/mark/openam]:
Debug Directory [/path/to/openam-tools/admin/debug]:
Log Directory [/path/to/openam-tools/admin/log]:
The scripts are properly setup under directory:
/path/to/openam-tools/admin/openam
Debug directory is /path/to/openam-tools/admin/debug.
Log directory is /path/to/openam-tools/admin/log.
The version of this tools.zip is: 10.1.0-XPRESS (2012-December-09 02:58)
The version of your server instance is: OpenAM 10.1.0-XPRESS
(2012-December-09 02:48)
```

After setup, the tools are located under a directory named after the instance of OpenAM.

```
$ ls openam/bin/
ampassword amverifyarchive ssoadm
```

On Windows, these files are .bat scripts.

5. Check that **ssoadm** works properly.

```
$ umask 366
$ echo password > /tmp/pwd.txt
$ openam/bin/ssoadm list-servers -u amadmin -f /tmp/pwd.txt

http://openam.example.com:8080/openam
```

The **ssoadm** commands can also be run from **ssoadm.jsp** in OpenAM, for example at <http://openam.example.com:8080/openam/ssoadm.jsp>, once the page has been enabled as described in the section on OpenAM **ssoadm.jsp** in the *Administration Guide* in the *Administration Guide*.

Not all the commands are available on **ssoadm.jsp**, however this limitation is not present for the command line tool.

6. (Optional) If you connect to OpenAM over SSL (HTTPS), and the SSL certificate was not signed by a CA whose certificate is found in the Java cacerts truststore (for example, you used a self-signed certificate as described in the *Administration Guide* procedure, *To Set Up OpenAM With HTTPS on Tomcat* in the *Administration Guide*), then edit the **ssoadm** (**ssoadm.bat** on Windows) script such that **ssoadm** recognizes the certificate.

Add two additional options to the **java** command in the script to identify the proper truststore and truststore password, depending on how you set up SSL.

```
-D"javax.net.ssl.trustStore=/path/to/tomcat/conf/keystore.jks"
-D"javax.net.ssl.trustStorePassword=changeit"
```

7. (Optional) If you have deployed OpenAM in a site configuration, edit the **ssoadm** (**ssoadm.bat** on Windows) script to map the site name to servers.

You add a **com.ipplanet.am.naming.map.site.to.server** property to the **java** command in the script.

```
-D"com.ipplanet.am.naming.map.site.to.server=
lb-url=openam-url[,lb-url=openam-url ...]"
```

Notice that the mapping is a comma separated list of server URLs, *openam-url*, in the site with the load balancer URL, *lb-url*. For example, if your site is behind <https://lb.example.com:443/openam>, and the OpenAM servers are at <http://openam1.example.com:8080/openam> and <http://openam2.example.com:8080/openam>, then you add the following property to the **java** command (without line breaks).

```
-D"com.iplanet.am.naming.map.site.to.server=  
https://lb.example.com:443/openam=http://openam1.example.com:8080/openam,  
https://lb.example.com:443/openam=http://openam2.example.com:8080/openam"
```

Procedure 3.2. To Set Up Configuration Tools

1. Make sure the **JAVA_HOME** environment variable is properly set.

```
$ echo $JAVA_HOME  
/path/to/jdk1.6
```

2. Unpack the tools from where you unzipped OpenAM.

```
$ cd /path/to/openam-tools/config  
$ unzip ~/Downloads/openam/  
openam-distribution-ssoadmintools-10.1.0-Xpress.zip  
Archive:  ~/Downloads/openam/openam-distribution-ssoadmintools-10.1.0-Xpress.zip  
  inflating: ssoadm.template  
  inflating: ssoadm.bat.template  
  inflating: ampassword.template  
  inflating: ampassword.bat.template  
  inflating: amverifyarchive.template
```

Set up configuration files based on the [sampleconfiguration](#) example, and then apply the configuration to a deployed OpenAM .war file using the following command.

```
$ java -jar configurator.jar -f config.file
```

The *config.file* is set up by default to use the embedded data store with OpenAM installed on [server1.example.com](#). You must edit the file before using it, as described in the *OpenAM Reference* for **configurator.jar** in the *Reference*.

Procedure 3.3. To Set Up Diagnostic Tool

The diagnostic tool, **ssodtool.sh** (**ssodtool.bat** on Windows), works in both graphical and console mode.

1. Make sure the **JAVA_HOME** environment variable is properly set.

```
$ echo $JAVA_HOME  
/path/to/jdk1.6
```

2. Unpack the tools from where you unzipped OpenAM.

```
$ cd /path/to/openam-tools/diagnostic
$ unzip ~/Downloads/openam/openam-distribution-ssoadmintools-10.1.0-Xpress.zip
...
  inflating: services/webcontainer/service.xml
  inflating: ssodtool.bat
  inflating: ssodtool.sh
```

You can start the graphical user interface by using the tools without options, or in console mode using the **ssodtool.sh --console** command.

Chapter 4

Installing OpenAM Distributed Authentication

When you need to prevent end users from having direct access to the service URLs OpenAM uses to manage access, you can deploy the distributed authentication service (DAS) in your DMZ, leaving OpenAM core services behind the firewall that end users cannot access.

You complete the following stages in deploying the DAS web service.

1. Make sure the cookie domain for the DAS is configured in OpenAM under Configuration > System > Platform.
2. Make sure the realms used have a Realm/DNS alias for the DAS configured in OpenAM under Access Control > *Realm Name* > General.
3. Deploy the `openam-distauth-10.1.0-Xpress.war` file into your web application container.

How you deploy the DAS `.war` file depends on your web application container. The procedure in this section shows how to deploy on Apache Tomcat.

4. Configure the DAS UI to access OpenAM core services.
5. Configure your firewall to allow end user access to the DAS.

Firewall configuration is not described here.

Procedure 4.1. To Deploy the DAS on Tomcat

1. Copy the `openam-distauth-10.1.0-Xpress.war` file into the `webapps/` directory.

```
cp ~/Downloads/openam/  
openam-distauth-10.1.0-Xpress.war /path/to/tomcat/webapps
```

2. Check that you see the initial DAS configuration screen in your browser.

Procedure 4.2. To Configure the DAS

1. Configure the DAS using the agent profile to connect to OpenAM.

Configuring DistAuth Application

Please provide the OpenSSO Server Information.

Server Protocol:	<input type="text" value="http"/>
Server Host:	<input type="text" value="openam.example.com"/>
Server Port:	<input type="text" value="8080"/>
Server Deployment URI:	<input type="text" value="/openam"/>
DistAuth Server Protocol:	<input type="text" value="http"/>
DistAuth Server Host:	<input type="text" value="openam.example.com"/>
DistAuth Server Port:	<input type="text" value="8080"/>
DistAuth Server Deployment URI:	<input type="text" value="/das"/>
DistAuth Cookie Name:	<input type="text" value="AMDistAuthCookie"/>
OpenAM LB Cookie Name:	<input type="text" value="ambcookie"/>
DistAuth LB Cookie Name:	<input type="text" value="DistAuthLBCookieName"/>
DistAuth LB Cookie Value:	<input type="text" value="DistAuthLBCookieValue"/>
Debug directory	<input type="text" value="/home/openam/das/debug"/>
Debug level	<input type="text" value="error"/>
Encryption Key	<input type="text" value="6Qq4oYSKwzqRZAIvhCdwhY0h7c7xekhw"/>
Application user name	<input type="text" value="UrlAccessAgent"/>
Application user password	<input type="password" value="....."/>
Confirm Application user password	<input type="password" value="....."/>

Most DAS configuration choices require no clarification. Hints for equivocal parameters follow.

Debug Level

Default is `error`. Other options include `error`, `warning`, `message`, and `off`.

Encryption Key

Do not change the password encryption key.

Application User Name

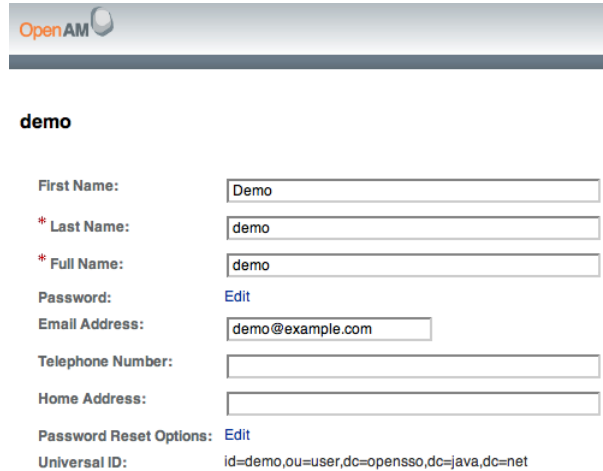
The DAS uses this identity, such as `UrlAccessAgent`, to authenticate to OpenAM.

Application User Password

The DAS uses this password to authenticate to OpenAM.

2. Login through the DAS to access OpenAM services.

For testing, you can login as user `demo`, password `changeit`.



The screenshot shows the OpenAM user profile page for the user 'demo'. The page includes the OpenAM logo at the top. Below the user name, there are several form fields for user information: First Name (Demo), Last Name (demo), Full Name (demo), Password (with an Edit link), Email Address (demo@example.com), Telephone Number, and Home Address. At the bottom, there are Password Reset Options (with an Edit link) and a Universal ID (id=demo,ou=user,dc=opensso,dc=java,dc=net).

When the `/openam/idm/EndUser` page is inside the firewall, and therefore not visible to users outside, redirect the browser after successful login to a page that exists. One way to do this is to use the `goto` parameter in the URL.

```
https://das.example.com/das/UI/Login?goto=absolute-successful-redirect-URL
```

On successful login, your browser stores an `AMDistAuthConfig` cookie that identifies the DAS.

3. Restrict permissions to the configuration for the DAS under the `$HOME/FAMDistAuth` directory of the user who runs the web container where you deployed the service.

The configuration file name ends in `AMDistAuthConfig.properties`.

If you deploy multiple DAS servers, you can configure them to forward requests to each other based on the `AMDistAuthConfig` cookie by setting the `com.sun.identity.distauth.cluster` property in this file. The following example lines are wrapped to fit on the page, but you put the entire property on a single line in the configuration file.

```
com.sun.identity.distauth.cluster=
http://das.example.com:8080/das/UI/Login,
http://das2.example.com:8080/das/UI/Login
```


Chapter 5

Installing OpenAM Client SDK Samples

Important

This samples mentioned in this chapter are not available in the current release.

The full download of OpenAM comes with a Java Client SDK and samples located in `samples/opensso-client.zip` where you unpacked the `openam_version.zip` file. The *Developer's Guide* in the *Developer's Guide* describes how to use the Java SDK, and the *OpenAM Java SDK API Specification* provides a reference to the public APIs.

To install the Client SDK, unzip `opensso-client.zip` where you want to install the SDK and examples.

```
$ cd samples/ ; unzip -oq opensso-client.zip
```

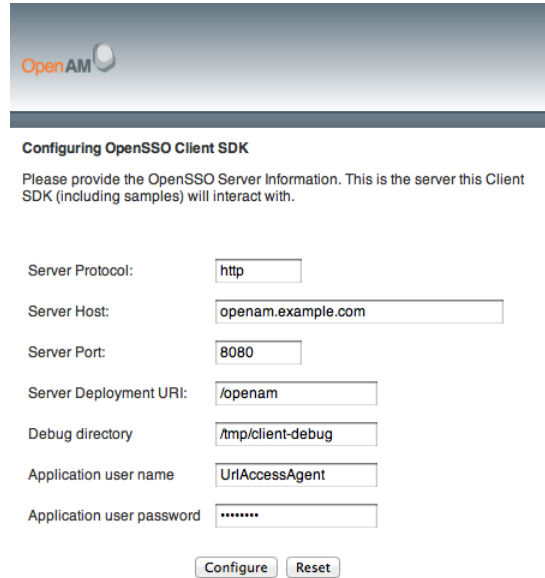
Procedure 5.1. To Deploy the Sample Web Application

The sample web application deploys in your container to show you the client SDK samples in action.

1. Deploy the `.war` in your Java web application container such as Apache Tomcat or JBoss.

```
$ cp war/opensso-client-jdk15.war /path/to/tomcat/webapps/client.war
```

2. Browse to the location where you deployed the client, and configure the application to access OpenAM using the application user name, `UrlAccessAgent`, and password configured when you set up OpenAM.



Configuring OpenSSO Client SDK

Please provide the OpenSSO Server Information. This is the server this Client SDK (including samples) will interact with.

Server Protocol:

Server Host:

Server Port:

Server Deployment URI:

Debug directory:

Application user name:

Application user password:

The sample client writes configuration information under `$HOME/OpenSSOClient/`, where `$HOME` is that of the user running the web application container.

3. Verify that you have properly configured the sample web application.
 - a. On the Client Samples page, click Access Management Samples.
 - b. In another browser tab page of the same browser instance, login to OpenAM as the OpenAM Administrator, `amadmin`.

This signs you into OpenAM, storing the cookie in your browser.

- c. On the Samples tab page, click the link under Single Sign On Token Verification Servlet.

If the sample web application is properly configured, you should see something like the following text in your browser.

```
SSOToken host name: 192.168.56.1
SSOToken Principal name: id=amadmin,ou=user,o=openam
Authentication type used: DataStore
IPAddress of the host: 192.168.56.1
SSO Token validation test succeeded
The token id is AQIC5wM2LY4SfcyeA2UgS0dLJIb-xjJClrk_EIXBpdzh7RI.*AAJTSQACMDE.*
Property: Company: Sun Microsystems
Property: Country: USA
User Attributes: {sunIdentityMSISDNNumber=[], mail=[],
dn=[uid=amAdmin,ou=people,o=openam], givenName=[amAdmin],
inetUserStatus=[Active], telephoneNumber=[], sn=[amAdmin],
roles=[Top-level Admin Role], employeeNumber=[], postalAddress=[],
iplanet-am-user-success-url=[], iplanet-am-user-failure-url=[],
cn=[amAdmin], iplanet-am-user-alias-list=[]}
```

Procedure 5.2. To Build the Command-Line Sample Applications

1. Compile the sample applications.

```
$ cd sdk/
$ sh scripts/compile-samples.sh
```

2. Configure the samples to access OpenAM.

```
$ sh scripts/setup.sh
Debug directory (make sure this directory exists): /tmp
Application user (e.g. URLAccessAgent) password: secret12
Protocol of the server: http
Host name of the server: openam.example.com
Port of the server: 8080
Server's deployment URI: /openam
Naming URL (hit enter to accept default value,
http://openam.example.com:8080/openam/namingservice):
$
```

3. Verify that you have properly configured the samples.

```
$ sh scripts/Login.sh
Realm (e.g. /): /
Login module name (e.g. DataStore or LDAP): DataStore
Login locale (e.g. en_US or fr_FR): fr_FR
DataStore: Obtained login context
Nom d'utilisateur?:demo
Mot de passe?:changeit
Login succeeded.
Logged Out!!
```

Chapter 6

Customizing the OpenAM End User Pages

When you deploy OpenAM to protect your web-based applications, users can be redirected to OpenAM pages for login and logout. ForgeRock provides pages localized for English, French, German, Spanish, Japanese, Korean, Simplified Chinese, and Traditional Chinese, but you might require additional language support for your organization.

Also, by default the end user pages have ForgeRock styling and branding. You likely want to change at least the images to reflect your organization. You might want to have different page customizations for different realms as well. This chapter address how to get started customizing OpenAM end user pages for your organizations and supported locales.

First you copy the pages to customize to the proper location, then you customize the files themselves.

Images in this example are located in `/path/to/tomcat/webapps/openam/images/`, and CSS in `/path/to/tomcat/webapps/openam/css/`. If you choose to modify images for your deployment, you can maintain the sizes to avoid having to customize page layout extensively.

Tip

The procedures below describe how to update a deployed version of OpenAM, so that you can see your changes without redeploying the application. This approach works for development as long as your web container does not overwrite changes. When developing with a web container that deploys OpenAM in a temporary location, such as JBoss or Jetty, restarting the container can overwrite your changes with the deployable `.war` content. For those web containers, you should also prepare a deployable `.war` containing your changes, and redeploy that file to check your work.

For production deployment you must package your changes in a custom OpenAM deployable `.war` file. To create a deployable `.war`, unpack the OpenAM `.war` file from `~/Downloads/openam` into a staging directory, apply your changes in the staging directory, and use the `jar` command to prepare the deployable `.war`.

Procedure 6.1. To Copy the Pages to Customize For the Top-Level Realm

Rather than changing the default pages, customize your own copy.

1. Change to the `config/auth` directory where you deployed OpenAM.

```
$ cd /path/to/tomcat/webapps/openam/config/auth
```

2. Copy the default files and optionally the localized files to `suffix[_locale]/html`, where `suffix` is the value of the RDN of the configuration suffix, such as `openam` if you use the default configuration

suffix `dc=openam,dc=forgerock,dc=org`, and the optional *locale* is, for example, `jp` for Japanese, or `zh_CN` for Simplified Chinese.

The following example copies the files for the Top-Level Realm (`/`) for a custom French locale.

```
$ mkdir -p openam/html
$ cp -r default/* openam/html
$ mkdir -p openam_fr/html
$ cp -r default_fr/* openam_fr/html
```

See *How OpenAM Looks Up UI files* for details.

Procedure 6.2. To Copy the Pages to Customize For Another Realm

As for the top-level realm, customize your copy rather than the default pages.

1. Change to the `config/auth` directory where you deployed OpenAM.

```
$ cd /path/to/tomcat/webapps/openam/config/auth
```

2. Depending on which locale you want to customize, copy the default files and optionally the localized files to `suffix[_locale]/services/realm/html`.

The following example copies the files for a custom French locale and a realm named `ventes`.

```
$ mkdir -p openam/html/ventes/html
$ cp -r default/* openam/services/ventes/html
$ mkdir -p openam_fr/services/ventes/html
$ cp -r default_fr/* openam_fr/services/ventes/html
```

Procedure 6.3. To Customize Files You Copied

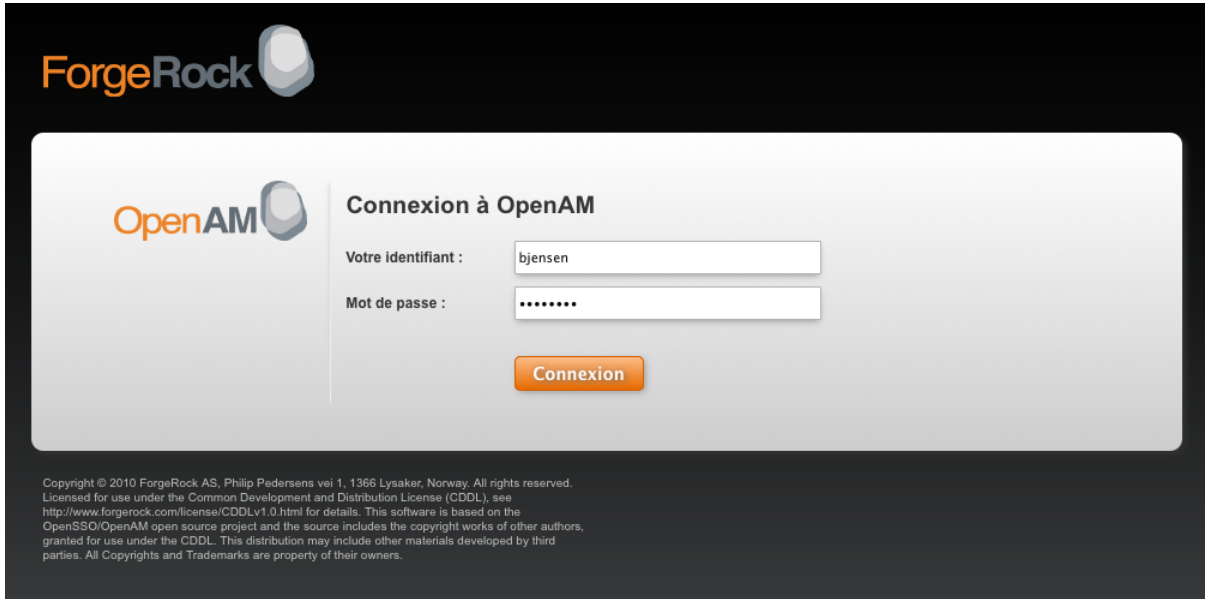
The `.jsp` files from the `default/` directory reference the images used in the OpenAM pages, and retrieve localized text from the `.xml` files. Thus you customize appearance through the `.jsp` files, being careful not to change the functionality itself. You customize the localized text through the `.xml` files.

1. Modify appearance if you must by editing the `.jsp`, image, and CSS files without changing any of the JSP tags that govern how the pages work.
2. Modify the localized text, using UTF-8 without escaped characters, by changing only the original text strings in the `.xml` files.

For example, to change the text in the default OpenAM login screen in the top-level realm for the French locale, edit `openam_fr/html/DataStore.xml`.

3. If necessary, restart OpenAM or the web container to test the changes you have made.

The following screen shot shows a customized French login page where the string `Nom d'utilisateur` has been replaced with the string `Votre identifiant`, and the string `Mot de passe` has been replaced with the string `Votre mot de passe` in `openam_fr/html/DataStore.xml`.



6.1. How OpenAM Looks Up UI Files

This section provides a more complete description of how OpenAM looks up UI files.

OpenAM uses the following information to look up the UI files.

Configuration suffix RDN

When you set up the OpenAM to store its configuration in a directory server, you provide the distinguished name of the configuration suffix, by default `dc=openam,dc=forgerock,dc=org`, therefore, the relative distinguished name attribute value is `openam`.

Client (browser) locale language

The client can specify a locale, which can consist of both a language and a territory, such as `en_GB` for British English. The language in this case is `en`.

Client (browser) locale territory

If the client local is `en_GB`, then the territory in this case is `GB`.

Platform locale language

The platform locale, defined for the platform where OpenAM runs, can also consist of both a language and a territory, such as `hu_HU`. In this example the platform locale language is `hu` for Hungarian.

Platform locale territory

If the platform locale is `hu_HU`, the platform locale territory is `HU` for Hungary.

Realm

Realms can be nested. OpenAM uses the nesting as necessary to look for files specific to a sub-realm before looking in the parent realm.

For all realms below the top level realm, OpenAM adds a `services` directory before the realm to the search path.

Client name

Client names identify the type of client. The default, `html`, is the only client name used unless client detection mode is enabled. When client detection mode is enabled, the client name can be different for mobile clients, for example.

File name

File names are not themselves localized. Thus `Login.jsp` has the same name for all locales, for example.

OpenAM tries first to find the most specific file for the realm and local requested, gradually falling back on less specific alternatives, then on other locales. The first and most specific location as follows.

```
suffix_client-locale-language_client-locale-territory/services/realm/client-name/file-name
```

Example 6.1. UI File Lookup

OpenAM looks up `Login.jsp` in the following order for a realm named `realm`, with the browser requesting `en_GB` locale, the platform locale being `hu_HU`, and the configuration suffix named `o=openam`. The client name used in this example is the generic client name `html`.

```
openam_en_GB/services/realm/html/Login.jsp
openam_en_GB/services/realm/Login.jsp
openam_en_GB/services/html/Login.jsp
openam_en_GB/services/Login.jsp
openam_en_GB/html/Login.jsp
openam_en_GB/Login.jsp
openam_en/services/realm/html/Login.jsp
openam_en/services/realm/Login.jsp
openam_en/services/html/Login.jsp
openam_en/services/Login.jsp
openam_en/html/Login.jsp
openam_en/Login.jsp
openam_hu_HU/services/realm/html/Login.jsp
openam_hu_HU/services/realm/Login.jsp
openam_hu_HU/services/html/Login.jsp
openam_hu_HU/services/Login.jsp
openam_hu_HU/html/Login.jsp
```

```
openam_hu_HU/Login.jsp
openam_hu/services/realm/html/Login.jsp
openam_hu/services/realm/Login.jsp
openam_hu/services/html/Login.jsp
openam_hu/services/Login.jsp
openam_hu/html/Login.jsp
openam_hu/Login.jsp
openam/services/realm/html/Login.jsp
openam/services/realm/Login.jsp
openam/services/html/Login.jsp
openam/services/Login.jsp
openam/html/Login.jsp
openam/Login.jsp
default_en_GB/services/realm/html/Login.jsp
default_en_GB/services/realm/Login.jsp
default_en_GB/services/html/Login.jsp
default_en_GB/services/Login.jsp
default_en_GB/html/Login.jsp
default_en_GB/Login.jsp
default_en/services/realm/html/Login.jsp
default_en/services/realm/Login.jsp
default_en/services/html/Login.jsp
default_en/services/Login.jsp
default_en/html/Login.jsp
default_en/Login.jsp
default_hu_HU/services/realm/html/Login.jsp
default_hu_HU/services/realm/Login.jsp
default_hu_HU/services/html/Login.jsp
default_hu_HU/services/Login.jsp
default_hu_HU/html/Login.jsp
default_hu_HU/Login.jsp
default_hu/services/realm/html/Login.jsp
default_hu/services/realm/Login.jsp
default_hu/services/html/Login.jsp
default_hu/services/Login.jsp
default_hu/html/Login.jsp
default_hu/Login.jsp
default/services/realm/html/Login.jsp
default/services/realm/Login.jsp
default/services/html/Login.jsp
default/services/Login.jsp
default/html/Login.jsp
default/Login.jsp
```


Chapter 7

Setting Up OpenAM Session Failover

This chapter covers setting up session failover (SFO) when using multiple instances of OpenAM in a site configuration for high availability. The basic idea followed here is that you configure load balancing to be sticky, based on the value of an OpenAM cookie, `amlbcookie`, different for each OpenAM server. Should that server become unavailable, the load balancer fails client requests over to another OpenAM server. The other OpenAM server must then fail over the user session associated with the client.

SFO uses a highly available data store for OpenAM session data, shared by OpenAM servers in a site configuration. When the OpenAM server where a user authenticated goes down, other OpenAM servers can read the user session information from the highly available store, so the user does not have to authenticate again. When the original OpenAM server becomes available again, it can also read session information from the store, and carry on serving users with active sessions.

Configuring an Additional Instance provides the steps to configure each instance of OpenAM. The following procedure describes how to configure an OpenAM server to use a load balancer for the first instance if it was not part of the initial configuration.

Note

Following an upgrade to OpenAM 10.1.0 XPress release, all users will need to login again because OpenAM no longer uses Open Message Queue or Berkeley DB Java Edition for SFO.

Note

SFO is supported within a site or data center with a shared local area network. SFO is not supported across sites and data centers linked by wide area networks (WAN).

Procedure 7.1. Configure First OpenAM Instance Behind a Load Balancer

Before you set up SFO, first configure OpenAM in a site configuration with a load balancer as the entry point to the site. However, you may already have a working instance before realizing that multiple instances are necessary. The following steps walk you through setting up the load balancer for the first instance.

1. In the OpenAM console for the first instance of the site, select **Configuration > Servers and Sites** and click on the first server under **Servers**.
2. Under site **Site** click on **Parent Site** and select the load balancer.

3. Click on **Save**.

Procedure 7.2. Setting a Load Balancer on Existing Configurations

If you did not set up the site during initial configuration, then follow all the steps below.

1. In the OpenAM console for one of the servers in the site, select Configuration > Servers and Sites > Sites > New..., and then create a new site.

The site URL is the URL to the load balancer in front of your OpenAM servers in the site. For example, if your load balancer listens on host `lb.example.com` and port `8080`, with OpenAM under `/openam`, then your site URL is `http://lb.example.com:8080/openam`.

2. For each OpenAM server in the site, select Configuration > Servers and Sites > Servers > *Server Name*, and then set Parent Site to the site you created before saving your work.
3. (Optional) If you want to use sticky load balancing, configure your load balancer to inspect the value of the `amlbcookie` to determine which OpenAM server should receive the client request.

As your load balancer depends on the `amlbcookie` value, on each OpenAM server console in the site, select Configuration > Servers and Sites > Servers > *Server Name* > Advanced, makes sure that `com.iplanet.am.lbcookie.value` is unique. By default the value of the `amlbcookie` is set to the server ID for the OpenAM instance.

Note

When using SSL, the approach requires that you either terminate SSL on the load balancer, or that you re-encrypt traffic from the load balancer to the OpenAM servers.

If you must change `amlbcookie` values to make them unique, then your changes take effect after you restart the OpenAM server. (To check, login to the console and check the cookie value in your browser.)

4. Restart each OpenAM server or the web containers where the OpenAM servers run so that all configuration changes take effect.

Chapter 8

Removing OpenAM Software

This chapter shows you how to uninstall OpenAM core software. See the *Policy Agent Installation Guide* for instructions on removing OpenAM agents.

Procedure 8.1. To Remove OpenAM Core Software

After you have deployed and configured OpenAM core services, you have at least two, perhaps three or four, locations where OpenAM files are stored on your system.

You remove the internal OpenAM configuration store when you follow the procedure below. If you used an external configuration store, you can remove OpenAM configuration data after removing all the software.

1. Shut down the web application container in which you deployed OpenAM.

```
$ /etc/init.d/tomcat stop
Password:
Using CATALINA_BASE:   /path/to/tomcat
Using CATALINA_HOME:   /path/to/tomcat
Using CATALINA_TMPDIR: /path/to/tomcat/temp
Using JRE_HOME:        /path/to/jdk1.6/jre
Using CLASSPATH:       /path/to/tomcat/bin/bootstrap.jar:
                        /path/to/tomcat/bin/tomcat-juli.jar
```

2. Unconfigure OpenAM by removing configuration files found in the \$HOME directory of the user running the web application container.

For a full install of OpenAM core services, configuration files include the following.

- The configuration directory, by default `$HOME/openam`. If you did not use the default configuration location, then check in the OpenAM console under Configuration > Servers and Sites > *Server Name* > General > System > Base installation directory.
- The hidden file that points to the configuration directory.

For example, if you are using Apache Tomcat as the web container, this file could be `$HOME/.openamcfg/AMConfig_path_to_tomcat_webapps_openam_` OR `$HOME/.openssocfg/AMConfig_path_to_tomcat_webapps_openam_`.

```
$ rm -rf $HOME/openam $HOME/.openamcfg
```

or

```
$ rm -rf $HOME/openam $HOME/.openssocfg
```

Note

At this point, you can restart the web container and configure OpenAM anew if you only want to start over with a clean configuration rather than removing OpenAM completely.

If you used an external configuration store you must also remove the configuration manually from your external directory server. The default base DN for the OpenAM configuration is `dc=openam,dc=forgerock,dc=org`.

3. Undeploy the OpenAM web application.

For example, if you are using Apache Tomcat as the web container, remove the `.war` file and expanded web application from the container.

```
$ cd /path/to/tomcat/webapps/  
$ rm -rf openam.war openam/
```

4. If you have stored a download or unpacked version of OpenAM software on your system, you can now remove the files.

If you cannot find the original `.zip`, search for files named `openam-*.zip`.

```
$ find . -name "openam-*.zip"  
./Downloads/openam  
$ rm ./Downloads/openam
```

Index

C

Custom end user pages, 39

D

Downloading OpenAM, 3

I

Installing

- Behind the firewall, 33

- Full install, 12

- Interactive configuration, 17

- Java SDK samples, 36

- Load Balancer, 44

- No console, 12

- Session failover, 44

- Silent configuration, 31

- Starting over, 16

- Tools (ssoadm, etc.), 29

M

Memory requirements, 1

P

Prerequisites, 1

S

Software requirements, 1

U

Uninstalling, 46