



Installation Guide

OpenAM 10

Mark Craig

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. Installing OpenAM Core Services	1
1.1. Preparing Prerequisite Software	1
1.2. Obtaining OpenAM Software	3
1.3. Install the OpenAM Web Application	4
2. Installing OpenAM Tools	13
3. Installing OpenAM Core Only	17
4. Installing OpenAM Distributed Authentication	18
5. Installing OpenAM Client SDK Samples	22
6. Customizing the OpenAM End User Pages	25
6.1. How OpenAM Looks Up UI Files	27
7. Setting Up OpenAM Session Failover	30
8. Upgrading OpenAM Core Services	34
9. Removing OpenAM Software	38
Index	40

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

Installing OpenAM Core Services

This chapter covers tasks required to install OpenAM core services, and to ensure they run properly. It includes instructions on preparing your application server to run OpenAM, preparing your identity repository to handle OpenAM identities, deploying component .war files, installing OpenAM administration tools, and performing post-installation configuration.

To manage access to web resources on other servers, you can install policy agents that provide tight integration with OpenAM. See the *Policy Agent Installation Guide* for instructions on installing OpenAM agents in supported web servers and Java EE application containers.

1.1. Preparing Prerequisite Software

OpenAM core services require the following prerequisite software installed before you begin OpenAM installation.

- A Java 6 runtime environment

```
$ java -version
java version "1.6.0_29"
Java(TM) SE Runtime Environment (build 1.6.0_29-b11-402-11D50)
Java HotSpot(TM) 64-Bit Server VM (build 20.4-b02-402, mixed mode)
```

- A supported application server as the deployment container

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

OpenAM core services require a minimum JVM heap size of 1 GB, and a permanent generation size of 256 MB. Apply the following JVM options before start your application server.

```
-Xmx1024m -XX:MaxPermSize=256m
```

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

OpenAM examples here 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.

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

- A supported identity repository for user identity data

See the *Release Notes* in the *Release Notes* for a list. If you plan to use OpenAM for evaluation or testing, definitely use the embedded LDAP server as a configuration store and identity repository. ForgeRock also recommends using the embedded LDAP server as the configuration store when you have four or fewer instances of OpenAM in production.

Examples here use OpenDJ as the identity repository. 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/opensj`, created with the OpenDJ **create-rc-script** command, manages the service at system startup and shutdown.

```
#!/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 "${1}" in
start)
  /bin/su mark -c "${INSTALL_ROOT}/bin/start-ds --quiet"
  exit $?
;;
stop)
  /bin/su mark -c "${INSTALL_ROOT}/bin/stop-ds --quiet"
  exit $?
;;
restart)
  /bin/su mark -c "${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](#), that you can download.

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

1.2. Obtaining OpenAM Software

Download OpenAM releases from <http://forgerock.com/downloads.html>. At the download location you find links to stable releases, nightly builds for testing, previous releases, and also OpenAM policy agent releases.

For each release of the OpenAM core services, you can download the entire package as a .zip archive, only the OpenAM .war file, only the administrative tools as a .zip archive, or only the OpenAM source code used to build the release.

After you unzip the archive of the entire package, you get an `opensso` directory including a README, a set of license files, and the following directories. See the *File Layout* reference in the *Reference* for details.

`deployable-war`

The deployable .war file as well as the tools and files required to create any specialized .war files you deploy.

`docs`

Javadoc API specifications for the public APIs exposed by OpenAM

fedlet

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

integrations

Resources for integrating OpenAM with third-party access and identity management software

ldif

Schema and index definitions for use with external LDAP identity repositories

libraries

Client SDK and policy agent libraries

patches

Location for patches to apply to OpenAM

samples

Sample source files demonstrating how to use the OpenAM client SDK

tools

OpenAM tools for managing SSO, configuring deployed .war files, patching deployed .war files, managing sessions, and diagnosing deployment issues

xml

OpenAM service and default delegation policy configuration files

1.3. Install the OpenAM Web Application

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

Procedure 1.1. To Deploy OpenAM on Tomcat

1. Copy the .war file to the directory where web applications are stored.

```
$ cp deployable-war/opensso.war /path/to/tomcat/webapps/openam.war
```

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`.



Copyright © 2010-2011 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 1.2. To Configure OpenAM With Defaults (For Testing)

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.

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.

OpenAM Configurator

Default Configuration Option

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

Default User [amAdmin]

* Password OK

* Confirm Password

Default Policy Agent [UriAccessAgent]

* Password OK

* Confirm Password

- 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 HELP

User: amAdmin Server: localhost.localdomain

OpenAM

Common Tasks | Access Control | Federation | Web Services | Configuration | Sessions

- Test Beta Console**
Access the beta OpenAM administration console to try the new Entitlements framework and new work flows to implement SAMLv2 Federation and Web Services Security. This beta console should only be used for testing purposes.
- Create SAMLv2 Providers**
Use these work flows to create hosted or remote identity and service providers for SAMLv2 Federation.
 -
 -
 -
 -
- 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.
- 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 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.
- Test Federation Connectivity**
Use this automated test to determine if federation connections are being made successfully, or to identify where issues might be located.
- Get Product Documentation**
Launch the OpenAM Resource Center to read the product documentation and other technical articles. Also, access links to general project information, FAQs, training and community blogs.

Procedure 1.3. 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/.openssocfg
```

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=opensso,dc=java,dc=net`).

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 1.4. 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 OK

* Confirm Password

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

OpenAM Configurator
✕

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 OK

* Cookie Domain

* Platform Locale

* Configuration Directory

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.

Alternatively, if you already manage an OpenDJ or DSEE deployment, you can choose to store OpenAM configuration data in your existing directory service.

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
 - Sun Java System Directory Server
 - Active Directory with Host and Port
 - Active Directory Application Mode
 - OpenDS/OpenDJ
 - AD with Domain Name
 - IBM Tivoli Directory Server
- * SSL/TLS Enabled
- * Directory Name
- * Port OK
- * Root Suffix OK
- * Login ID OK
- * 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 truststore 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, accept the defaults.

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

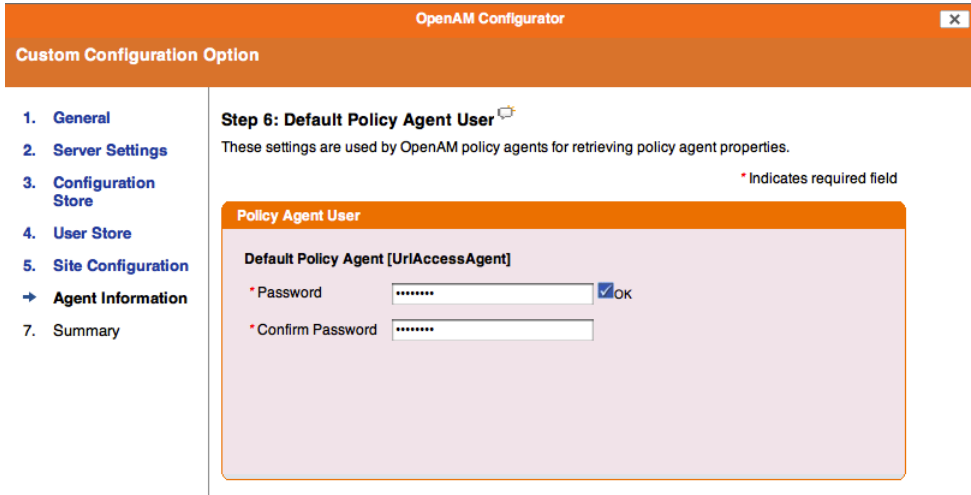
Site Configuration Details

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

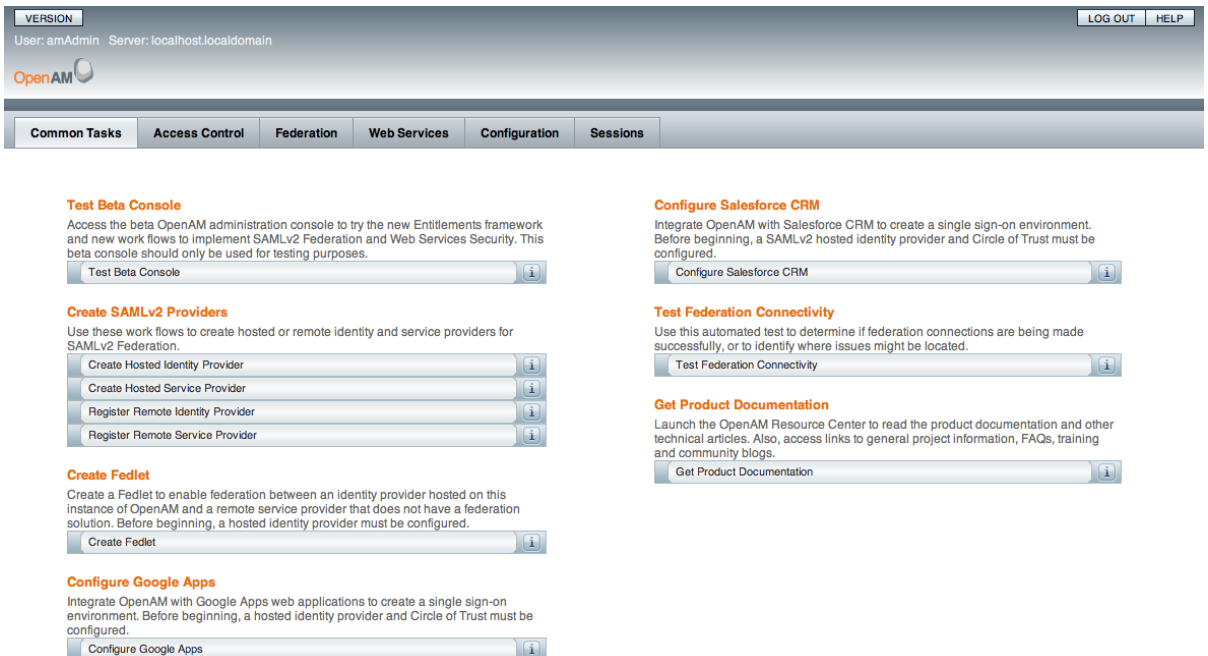
* Site Name

* Load Balancer URL

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



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



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

Chapter 2

Installing OpenAM Tools

OpenAM tools are found in the `tools/` directory where you unpacked the archive of the entire package.

README

Quick description of the tools .zip files. Each tools .zip also includes a specific README.

`tools/ssoAdminTools.zip`

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

`tools/ssoConfiguratorTools.zip`

Tools for command line installation, upgrade, and initial configuration

`tools/ssoDiagnosticTools.zip`

Tool to check for problems with your deployment and configuration

`tools/ssoSessionTools.zip`

Session failover utilities. Installation for these tools is described in the chapter on *Setting Up OpenAM Session Failover*.

Procedure 2.1. To Set Up Administration 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/admin/tools  
$ unzip /path/to/OpenAM/tools/ssoAdminTools.zip  
...  
  inflating: template/windows/bin/amverifyarchive.bat.template  
  inflating: template/windows/bin/ssoadm.bat.template
```

3. 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/admin/tools/debug]:
Log Directory [/path/to/admin/tools/log]:
The scripts are properly setup under directory: /path/to/admin/tools/openam
Debug directory is /path/to/admin/tools/debug.
Log directory is /path/to/admin/tools/log.
The version of this tools.zip is: (2011-July-11 00:05)
The version of your server instance is: (2011-July-11 00:05)
```

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.

4. 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.

5. (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"
```

6. (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 2.2. 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.

```
$ unzip /path/to/OpenAM/tools/ssoConfiguratorTools.zip  
Archive: /path/to/OpenAM/tools/ssoConfiguratorTools.zip  
  inflating: README.setup  
  inflating: configurator.jar  
  inflating: sampleconfiguration  
  inflating: sampleupgrade  
  inflating: upgrade.jar
```

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 2.3. 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.

```
$ unzip /path/to/OpenAM/tools/ssoDiagnosticTools.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 3

Installing OpenAM Core Only

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.

Procedure 3.1. To Create the Core Services `.war` File

1. Unpack the `opensso.war` file into a temporary directory.

```
$ mkdir -p /tmp/headless ; cd /tmp/headless
$ jar xf /path/to/OpenAM/deployable-war/opensso.war
```

2. Create the `headless.war` file.

```
$ cd /path/to/OpenAM/deployable-war
$ sh createwar.sh -s /tmp/headless --type noconsole -w headless.war
```

Procedure 3.2. To Deploy the Core Services On Tomcat

1. Put the `headless.war` you created in the Tomcat `webapps/` directory.

```
$ mv headless.war /path/to/tomcat/webapps/
```

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

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. Create the `.war` file for the DAS using OpenAM software.
4. Deploy the DAS `.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.

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

Firewall configuration is not described here.

Procedure 4.1. To Create the DAS `.war` File

1. Unpack the `opensso.war` file into a temporary directory.

```
$ mkdir -p /tmp/das ; cd /tmp/das
$ jar xf /path/to/OpenAM/deployable-war/opensso.war
```

2. Create the `das.war` file.

```
$ cd /path/to/OpenAM/deployable-war
$ sh createwar.sh --staging /tmp/das --type distauth --warfile das.war
```

Procedure 4.2. To Deploy the DAS on Tomcat

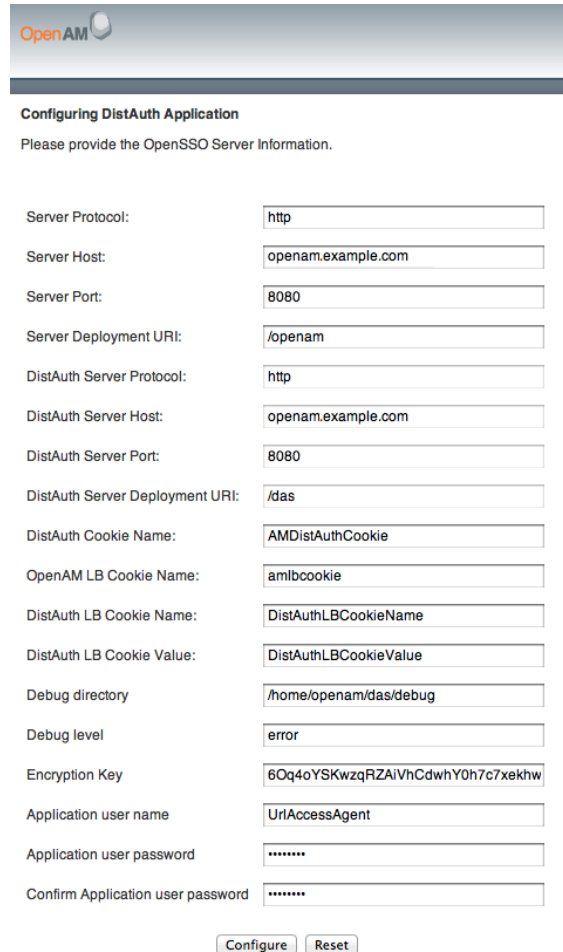
1. Put the `das.war` you created in the Tomcat `webapps/` directory.

```
$ mv das.war /path/to/tomcat/webapps/
```

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

Procedure 4.3. 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="amlbcookie"/>
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="6Oq4oYSKwzqRZAiVhCdwhY0h7c7xekhw"/>
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

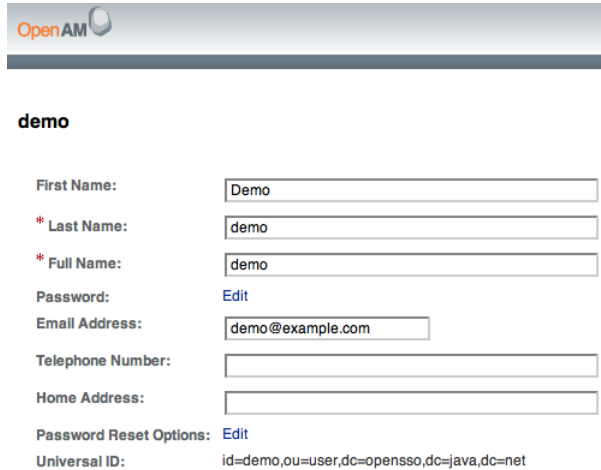
The DAS uses this identity, such as `UrlAccessAgent`, to authenticate to OpenAM.

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`.



demo

First Name:

* Last Name:

* Full Name:

Password: [Edit](#)

Email Address:

Telephone Number:

Home Address:

Password Reset Options: [Edit](#)

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

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
```

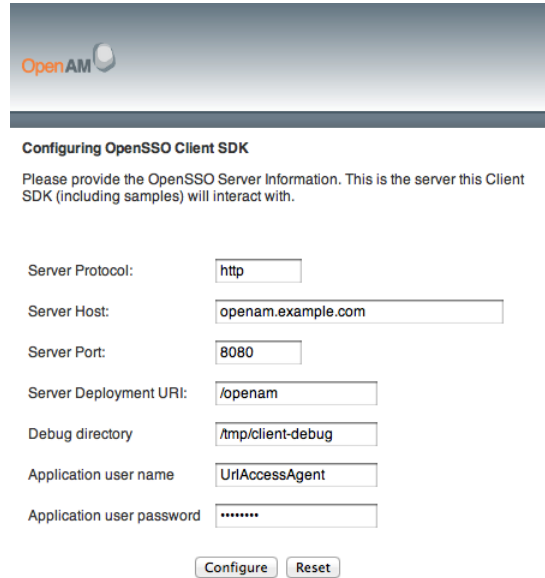
Procedure 5.1. To Deploy the Sample Web Application

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): LDAP
Login locale (e.g. en_US or fr_FR): en_US
LDAP: Obtained login context
User Name:bjensen
Password:hifalutin
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 `/path/to/OpenAM/deployable-war` into a staging directory, apply your changes in the staging directory, and use the `createwar.sh` or `createwar.bat` script 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 `opensso` if you use the default configuration

suffix `dc=opensso,dc=java,dc=net`, 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
$ 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

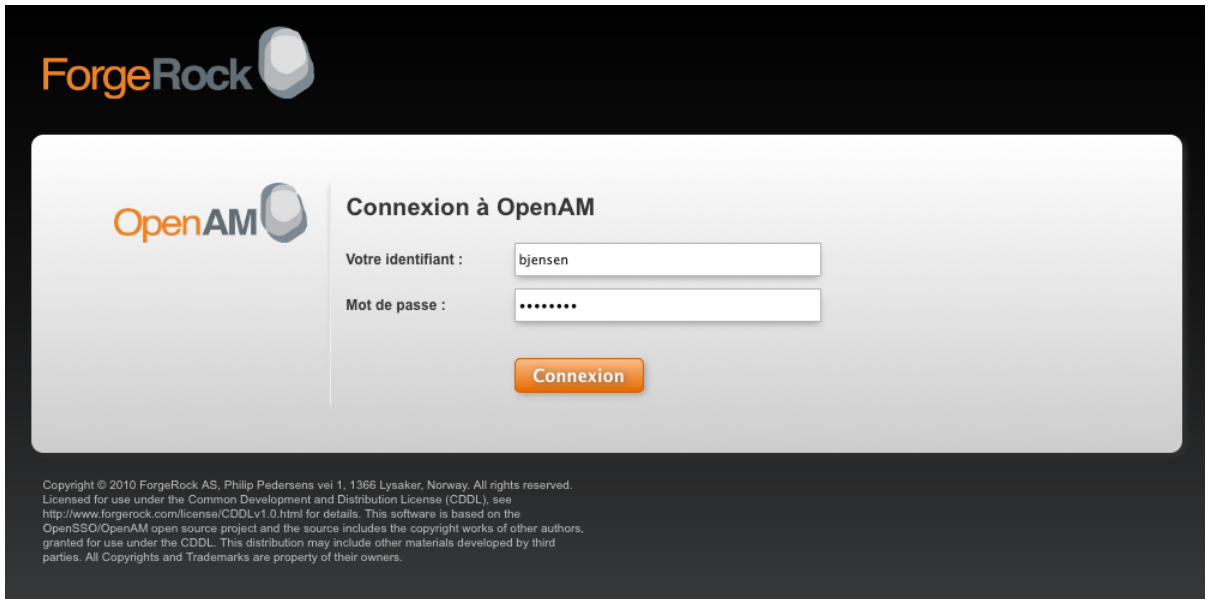
The `.jsp` files from the `default/` directory reference the images used in the OpenAM pages, and retrieve localized text from the `.xml` files. Thus you customize appearance through the `.jsp` files, being careful not to change the functionality itself. You customize the localized text through the `.xml` 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=openSSO,dc=java,dc=net`, therefore, the relative distinguished name attribute value is `openSSO`. If instead you set the configuration suffix to be `o=openam`, the RDN 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 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.

Session failover 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.

Note

Session failover is supported within a site or data center with a shared local area network. Session failover is not supported across sites and data centers linked by wide area networks (WAN). Latency over the WAN can cause issues with the underlying message queue, and therefore prevents reliable session failover.

Procedure 7.1. Before You Start

Before you set up session failover, first configure OpenAM in a site configuration with a load balancer as the entry point to the site. The most expedient way to configure the site is to set it up during initial OpenAM configuration, where OpenAM can manage and replicate server configuration for availability. 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.)

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

Procedure 7.2. To Prepare the Session Data Service

The session data service relies on Open Message Queue and Berkeley DB Java Edition. You set up the session failover service in a site cluster to serve as the highly available session data store.

- Install the session tools from `ssoSessionTools.zip` on at least two, and generally not more than four, servers where you run OpenAM.¹

For example, you can install the session tools in the OpenAM configuration directory.

```
$ cd $HOME/openam
$ unzip /path/to/OpenAM/tools/ssoSessionTools.zip
...
$ ./setup
Name of the directory to install the scripts (example: sfoscripts):sfoscripts
The scripts are properly setup under directory: /home/user/openam/sfoscripts
JMQ is properly setup under directory /home/user/openam/jmq
```

- Start the Message Queue broker in order to configure user accounts.

```
$ cd jmq/imq/bin
$ ./imqbrokerd -name aminstance -port 7777 &
```

- Change the default admin password from `admin` to something else.

```
$ ./imqusermgr update -u admin -p password -i aminstance
User repository for broker instance: aminstance
Are you sure you want to update user admin? (y/n)[n] y
User admin successfully updated.
```

¹You install more than one instance of the session tools in case an instance crashes and must fail over to another instance. At the same time, session failover requires that messages be sent across the network from one instance to another to stay in sync in case of failover. If you install too many instances, however, then the increase in network traffic for synchronization can impair performance.

4. Disable the default `guest` account.

```
$ ./imqusermgr update -u guest -a false -i aminstance
User repository for broker instance: aminstance
Are you sure you want to update user guest? (y/n)[n] y
User guest successfully updated.
```

5. Add a user for the session failover service.

```
$ ./imqusermgr add -u openamuser -g admin -p secret12 -i aminstance
User repository for broker instance: aminstance
User openamuser successfully added.
```

6. Create a password file for the session failover service.

```
$ cd ../../../../sfoscripts/bin/
$ ./amsfopassword --encrypt secret12 --passwordfile /home/user/openam/mqpwd.txt
os.name=Linux
SUCCESSFUL
```

7. Stop the broker you started on port 7777.

```
$ fg
./imqbrokerd -name aminstance -port 7777 (wd: ~/openam/jmq/imq/bin)
^C
[05/Mar/2012:08:35:22 CET] [B1047]: Shutting down broker...
[05/Mar/2012:08:35:22 CET] [B1077]: Broadcast good-bye to all connections ...
[05/Mar/2012:08:35:22 CET] [B1078]: Flushing good-bye messages ...
[05/Mar/2012:08:35:23 CET] [B1048]: Shutdown of broker complete.
```

8. Configure the session failover service for the site.

For each session tools installation, edit the `/home/user/openam/sfoscripts/config/lib/amsfo.conf` configuration file to change at least the `USER_NAME`, `CLUSTER_LIST`, and `PASSWORDFILE` parameters.

`USER_NAME` should match the user you created for the session failover service.

```
USER_NAME=openamuser
```

`CLUSTER_LIST` specifies the `host:port` combinations for all the session failover services you configure for the site.

```
CLUSTER_LIST=openam.example.com:7777,openam2.example.com:7777
```

`PASSWORDFILE` specifies the path to the password file you created.

```
PASSWORDFILE=/home/user/openam/mqpwd.txt
```

You can optionally set `AMSESSIONDB_ARGS="-v"` to log additional information.

Procedure 7.3. To Enable Session Failover

Enabling session failover at this point involves configuring OpenAM to use the session data store, and then starting services. Examples in this procedure show OpenAM running in Apache Tomcat.

1. On one of the OpenAM servers in the site, login to the console, and then select Configuration > Global > Session > Instances > New... to set up session data store access.

Provide a user name for the Session Store User, such as `openamuser`, with the same password you entered into the password file, and the Database Url is the `host:port` combination that you entered for the `CLUSTER_LIST` parameter.

Be sure to Add the new instance, and then also Save your configuration changes. The configuration changes take effect after you restart OpenAM.

2. Stop each OpenAM server in the site.

```
$ /etc/init.d/tomcat stop
```

3. Start each session data service in the cluster.

```
$ cd /home/user/openam/sfoscripts/bin
$ ./amsfo start
```

By default, the log and session data store are located under `/tmp/amsession/`.

```
$ tail -f /tmp/amsession/logs/amsessiondb.log
...
Initializing and connecting to the Message Queue server ...
Successfully started.
```

4. Start each OpenAM server in the site.

```
$ /etc/init.d/tomcat start
```

Wait for each OpenAM server to start before starting another.

```
$ tail -f /path/to/tomcat/logs/catalina.out
...
INFO: Server startup in 26047 ms
```

After OpenAM has started, you can test session failover.

Chapter 8

Upgrading OpenAM Core Services

This chapter shows you how to patch and to upgrade OpenAM core services. See the *Policy Agent Installation Guide* for instructions on upgrading OpenAM agents.

This chapter uses Apache Tomcat in examples. If you use a different web application container for OpenAM, adapt the examples to your environment.

Note

For complex and legacy deployments, ForgeRock can assist you through the upgrade process. Send mail to info@forgerock.com for more information.

Procedure 8.1. To Upgrade From OpenAM 9.5 Or Later

ForgeRock has considerably simplified OpenAM core services upgrade with respect to earlier versions. If you have already moved to OpenAM 9.5 or later, follow these steps.

1. If you have customized end user OpenAM files for your deployment, build a new .war with the new version of OpenAM that includes your customizations.
2. If OpenAM is running in a site deployment with multiple servers, configure your load balancer to take the OpenAM server out of the site pool during upgrade.
3. Stop the container where OpenAM is running.

```
$ /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
```

4. Make a copy of the OpenAM configuration directory that was created at installation time.

In the following example, the OpenAM configuration directory is `$HOME/openam`, where `$HOME` is the home directory of the user running the container where OpenAM is deployed.

```
$ cd $HOME
$ cp -r openam openam-orig
```

Files in the hidden `$HOME/.openssocfg/` directory are not changed during upgrade.

5. Make a backup copy of the OpenAM configuration stored in your configuration store.
6. Move the current OpenAM web application aside.

```
$ cd /path/to/tomcat/webapps
$ mv openam openam-orig ; mv openam.war openam.war.orig
```

7. Deploy the new OpenAM in place of the old.

```
$ cp /path/to/new/OpenAM/deployable-war/opensso.war openam.war
```

8. If your container caches OpenAM files, clear out the cache.

```
$ rm -rf /path/to/tomcat/work/Catalina/localhost/openam
```

9. Start the container to run the new OpenAM.

```
$ /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
```

10. In your browser, visit the location where OpenAM is deployed, such as <http://openam.example.com:8080/openam>, and then click Upgrade to OpenAM 10.0.0.



Upgrade Available

An older version of configuration has been found

Upgrade to OpenAM 10

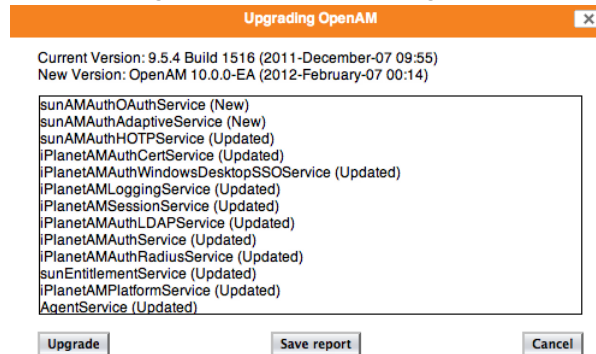
Please read the [release notes](#) before upgrading.

NOTE: you should NOT upgrade if you are running OpenAM version 9.0 or older.

[Upgrade to OpenAM 10](#)

Copyright © 2010-2011 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.

- In the Upgrading OpenAM screen, click Save Report to download an OpenAM Upgrade Report listing necessary configuration changes, and then click Upgrade.



Log messages pertaining to upgrade of the embedded directory are written to the web container log.

- When upgrade completes, restart OpenAM or the container where it runs, and continue your work.

In site deployments with multiple OpenAM servers using the embedded OpenDJ configuration store, you must keep the first upgraded server running when upgrading the others.

This allows secondary servers to correctly replicate their embedded configuration stores with that of the first server you upgraded.

- If OpenAM is running in a site deployment with multiple servers, configure your load balancer to return the OpenAM server to the site pool.

Tip

You can use the `upgrade.jar` utility, installed from `ssoConfiguratorTools.zip`, to perform the upgrade configuration after you deploy the new OpenAM `.war` file.

Instead of upgrading the new OpenAM deployment through OpenAM console as described in the procedure above, edit the `sampleupgrade` properties file next to `upgrade.jar` to set the `SERVER_URL` and `DEPLOYMENT_URI` for your environment, and then run the `upgrade.jar` utility.

```
$ java -jar upgrade.jar --file sampleupgrade
...
Upgrade Complete.
```

Procedure 8.2. To Revert From OpenAM Upgrade

If you must revert from an upgraded version of OpenAM, then the quickest way to return to the earlier version is to restore all the earlier files. You also must return to an earlier version of the configuration store data.

1. Stop OpenAM or the container where OpenAM is running.
2. Restore the old version of OpenAM files, including the .war and the configuration directory.
3. If your container caches OpenAM files, clear out the cache.

```
$ rm -rf /path/to/tomcat/work/Catalina/localhost/openam
```

4. Restore the earlier version of the configuration store data.
5. (Optional) If you run OpenAM in Tomcat, remove files from the working directory where Tomcat has stored JSP files from the upgraded version.

```
$ rm -fr /path/to/tomcat/work/Catalina/localhost/openam
```

If you skip this step, you can see an error like the following when logging into the console after reverting to the older version.

```
amConsole:10/29/2011 05:42:51:812 PM BST: Thread[http-8080-3,5,main]
ERROR: ConsoleServletBase.onUncaughtException
javax.servlet.ServletException: java.lang.NoClassDefFoundError:
  org/forgerock/openam/console/ui/taglib/header/CCHtmlHeaderTag
    at org.apache.jasper.runtime.PageContextImpl.doHandlePageException
      (PageContextImpl.java:862)
    at org.apache.jasper.runtime.PageContextImpl.handlePageException
      (PageContextImpl.java:791)
    at org.apache.jsp.console.task.Home_jsp._jspService(Home_jsp.java:564)
    at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
```

6. Start the earlier version of OpenAM or the container where OpenAM runs.

Chapter 9

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 9.1. 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/.openssocfg/AMConfig_path_to_tomcat_webapps_openam_`.

```
$ 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=opensso,dc=java,dc=net`.

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_nightly_20110808.zip  
$ rm ./Downloads/openam_nightly_20110808.zip
```

If you cannot find the unpacked version, you might search for the directory named `deployable-war`.

```
$ locate deployable-war/  
/path/to/OpenAM/deployable-war/README  
...  
$ rm -rf /path/to/OpenAM
```

Index

C

Custom end user pages, 25

D

Downloading OpenAM, 3

I

Installing

- Behind the firewall, 18

- Full install, 1

- Interactive configuration, 7

- Java SDK samples, 22

- No console, 17

- Session failover, 30

- Silent configuration, 15

- Starting over, 6

- Tools (ssoadm, etc.), 13

M

Memory requirements, 1

S

Software requirements, 1

U

Uninstalling, 38

Upgrading, 34