OpenAM Upgrade Guide

Version 13.5

Mark Craig
Gene Hirayama
Mike Jang

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

Abstract

This guide shows you how to upgrade OpenAM. OpenAM provides open source Authentication, Authorization, Entitlement, and Federation software.

Table of Contents

Preface

This guide describes how to upgrade OpenAM servers, policy agents, and tools.

1 Who Should Use this Guide

This guide is for anyone who needs to upgrade an OpenAM deployment. This guide assumes you are familiar with OpenAM installation and configuration, and that you are familiar with the current OpenAM deployment that you plan to upgrade.

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.

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. In the following example, the query string parameter `_prettyPrint=true` is omitted and some of the output is replaced with an ellipsis (`...`):

```
$ curl https://bjensen:hifalutin@opendj.example.com:8443/users/newuser
{
  "_rev" : "000000005b337348",
  "_id" : "newuser",
  ...
}
```

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 core documentation, such as this document, aims to be technically accurate and complete with respect to the software documented.

Core documentation therefore follows a three-phase review process designed to eliminate errors:

- Product managers and software architects review project documentation design with respect to the readers' software lifecycle needs.
- Subject matter experts review proposed documentation changes for technical accuracy and completeness with respect to the corresponding software.
- Quality experts validate implemented documentation changes for technical accuracy, completeness in scope, and usability for the readership.

The review process helps to ensure that documentation published for a ForgeRock release is technically accurate and complete.

Fully reviewed, published core documentation is available at http://backstage.forgerock.com/. Use this documentation when working with a ForgeRock Enterprise release.

You can find pre-release draft documentation at the online community resource center. Use this documentation when trying a nightly build.

4 Joining the ForgeRock Community

Visit the Community resource center where you can find information about each project, download nightly builds, browse the resource catalog, ask and answer questions on the forums, find community events near you, and of course get the source code as well.

5 Getting Support and Contacting ForgeRock

ForgeRock provides support services, professional services, classes through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. If you have any questions, contact ForgeRock using the address or telephone number nearest to you. Find the latest addresses and telephone numbers at https://www.forgerock.com, or send an email to ForgeRock at info@forgerock.com.

Chapter 1

About Upgrading OpenAM

This chapter covers common aspects of upgrading an OpenAM deployment, whether you are moving to a new maintenance release, upgrading to a new major release, or migrating from a legacy release to a newer OpenAM release.

Release levels, and how much change to expect in a maintenance, minor, or major release, are defined in Section A.1, "ForgeRock Product Release Levels" in the *OpenAM Administration Guide*. Release levels are identified by version number.

1.1 Supported Upgrade Paths

The following table contains information about the supported upgrade paths to OpenAM 13.5:

Table 1.1. Upgrade Paths

Version	Upgrade Supported?
OpenAM 9.0.x	No
OpenAM 9.5.x	No
OpenAM 10.0.x	No
OpenAM 11.0.x	Yes

Version	Upgrade Supported?
OpenAM 12.0.x	Yes
OpenAM 13.0.x	Yes

Note

Upgrading between OpenAM Enterprise and OpenAM OEM versions is not supported.

For more information, see Checking your product versions are supported in the *ForgeRock Knowledge Base*.

1.2 Planning the Upgrade

How much you must do to upgrade OpenAM software depends on the magnitude of the differences between the version you currently use and the new version.

- Maintenance releases have a limited effect on current functionality but contain necessary bug and security fixes. You should keep up to date with maintenance releases as the fixes are important and the risk of affecting service is minimal.
- When upgrading to a new major or minor release, always plan and test the changes before carrying out the upgrade in production. Make sure you read release notes for intervening versions with care, identifying any changes likely to affect your deployment, and then plan accordingly.
- These suggestions are true both for OpenAM server components, and also for policy agents.

To upgrade from an OpenAM server, use the Upgrade wizard. The OpenAM server Upgrade wizard appears when you replace a deployed OpenAM server `.war` file with a newer version and browse to the deployment URL. The Upgrade wizard brings the OpenAM configuration, including the version number, up to date with the new version. The CLI counterpart of the Upgrade wizard is openam-upgrade-tool-13.5.0.jar, which you install as described in Procedure 3.2, "To Set Up Configuration Tools" in the *OpenAM Installation Guide*.

1.3 Upgrading & Policies

When upgrading from OpenAM 11.0.x, the upgrade process changes how OpenAM represents policies. Most earlier policies transform directly to the newer representation.

If however the upgrade process encounters problems during the transformation, it writes messages about the problems in the upgrade log. When you open a policy in the policy editor that caused problems during the upgrade process, the policy editor shows the issues, but does not let you fix them directly. Instead you must create equivalent, corrected policies in order to use them in OpenAM.

You should therefore plan to test policy upgrade before upgrading the service, and to correct any problems encountered before using the upgraded service.

For details on how to configure OpenAM policies, see Chapter 3, "Defining Authorization Policies" in the *OpenAM Administration Guide*.

1.4 Best Practices for Upgrades

Be prepared before you begin an upgrade, even if the upgrade is for a maintenance release.

1.4.1 Route Around Servers During Downtime

Upgrading servers takes at least one of your OpenAM sites down while the server configurations are being brought up to date with the newer version. Plan for this site to be down, routing client applications to another site until the upgrade process is complete and you have validated the result. Make sure client application owners are well aware of the change, and let them know what to expect.

If you only have a single OpenAM site, make sure the downtime happens in a low usage window, and make sure you let client application owners plan accordingly.

During an upgrade you must restrict access to OpenAM Console: The Upgrade Wizard page does not require authorization; any user with access to OpenAM Console immediately after you deploy the new .war can therefore initiate the upgrade process.

1.4.2 Back Up the Deployment

Always back up your deployment before you upgrade, as you must be able to roll back should something go wrong during the upgrade process.

- Backing up your configuration as described in Chapter 21, "Backing Up and Restoring OpenAM Configurations" in the *OpenAM Administration Guide* is good for production environments.

- In preparation for upgrading OpenAM servers and their configurations, also take LDIF backups of the configuration store data in the directory servers. If possible, stop servers before upgrading and take a file system backup of the deployed servers and also of their configuration directories as well. This can make it easier to roll back from a failed upgrade.

  For example, if you deploy OpenAM server in Apache Tomcat under `/openam`, you might take a file system backup of the following directories for each OpenAM server.

  - `/path/to/tomcat/webapps/openam/`
  - `~/openam/`
  - `~/.openamcfg/`

- When upgrading web policy agents, take a file system backup of the policy agent installation and configuration directories.

  When upgrading Java EE policy agents, it can be easier to uninstall the new version and reinstall the old version than to restore from file system backup.

- When upgrading tools, keep copies of any tools scripts that you have edited for your deployment. Also back up any trust stores used to connect securely.

1.4.3 Apply Customization Before Upgrading

Before you upgrade OpenAM servers, prepare a .war file that contains any customizations you require.

Customizations include any changes you have made to the OpenAM server installation, such as the following.

- Plugins and extensions such as custom authentication modules, response attribute providers, post authentication plugins, SAML v2.0 attribute mappers, and OAuth 2.0 scope implementations

  These are described in the OpenAM Developer's Guide.

- Customized JSPs, redesigned login or service pages, additional CSS and visual content, and modified authentication module callback files

  These are described in the Chapter 5, "Customizing the OpenAM End User Pages" in the *OpenAM Installation Guide*.

- Any changes to OpenAM classes

- Any changes or additional Java class libraries (such as .jar files in `WEB-INF/lib`

1.4.4 Plan for Rollback

Sometimes even a well-planned upgrade operation fails to go smoothly. In such cases, you need a plan to roll back smoothly to the pre-upgrade version.

For OpenAM servers, you can roll back by restoring from file system backup. If you use an external configuration directory service, restore the old configuration from LDIF before restarting the old servers. For more information, see Chapter 21, "Backing Up and Restoring OpenAM Configurations" in the *OpenAM Administration Guide*.

For web policy agents, you can roll back by restoring from file system backup. If you used configuration only available to newer agents, restore the pre-upgrade configuration before restarting the old agents.

For Java EE policy agents, uninstall the newer agents and reinstall the older agents, including the old configurations.

Chapter 2
Upgrading OpenAM Servers

This chapter covers upgrade from OpenAM core server 11.0.0 or later to the current version. For other OpenAM components, see Chapter 4, "Upgrading OpenAM Components".

OpenAM server upgrade relies on the Upgrade Wizard to make the necessary changes to the configuration store. You must then restart OpenAM or the container in which it runs. Even a version number change requires that you run the Upgrade Wizard, so needing to run the Upgrade Wizard says nothing about the significance of the changes that have been made to OpenAM. You must run the Upgrade Wizard even for maintenance releases.

Make sure you try upgrading OpenAM in a test environment before applying the upgrade in your production environment.

Procedure 2.1. To Upgrade From a Supported OpenAM Version

Follow these steps to upgrade a site of OpenAM servers. For information on the versions of OpenAM are supported for upgrade, see Section 2.6, "Supported Upgrade Paths" in the *OpenAM Release Notes*.

During the upgrade process, you must take the OpenAM servers in the site out of production, instead redirecting client application traffic elsewhere. This is required because upgrade involves making changes to OpenAM's configuration model. If the upgrade fails, you must be able to roll back before the configuration changes impact other sites.

Important

Do *not* perform an upgrade by deploying the new version and then importing an existing configuration by running the **ssoadm import-svc-config** command. Importing an outdated configuration can result in a corrupted installation.

1. Prepare your customized OpenAM server `.war` file.
2. **Back up your deployment**.
3. Route client application traffic to another site during the upgrade.
4. For servers in the site, stop OpenAM, or if necessary stop the container where OpenAM runs.
5. For servers in the site, deploy your customized server .war file.

   When you deploy the new `.war` file, you might have to delete working files left by the old installation. For example, if you deploy on Apache Tomcat, replacing `/path/to/tomcat/webapps/openam.war`, then also recursively delete the `/path/to/tomcat/webapps/openam/` and `/path/to/tomcat/work/Catalina/localhost/openam/` directories before restarting the server.
6. For servers in the site, restart OpenAM or the container where it runs.
7. For the first server in the site, follow the instructions in the Upgrade Wizard.

   Alternatively for a silent, unattended upgrade, you can use the `openam-upgrade-tool-13.5.0.jar` tool to upgrade the server configuration in a command-line script.

   First you must install the tool. The procedure, Procedure 3.2, "To Set Up Configuration Tools" in the *OpenAM Installation Guide*, describes how to install the tool.

   The upgraded server must be deployed and running when you use the tool.

   The openam-upgrade-tool-13.5.0.jar relies on a properties file to upgrade OpenAM server.

```
$ cp sampleupgrade upgrade.properties
$ vi upgrade.properties
$ grep -v "^#" upgrade.properties
SERVER_URL=http://openam.example.com:8080
DEPLOYMENT_URI=/openam
ACCEPT_LICENSES=true
```

When the new OpenAM server `.war` file is deployed and running, you can upgrade by using the tool with the properties file.

```
$ java -jar openam-upgrade-tool-13.5.0.jar --file upgrade.properties

Writing Backup; Done.
Upgrading Services
New service iPlanetAMAuthPersistentCookieService; Done.
New service iPlanetAMAuthOpenIdConnectService; Done.
New service OAuth2Provider; Done.
New service iPlanetAMAuthDevicePrintModuleService; Done.
New service crestPolicyService; Done.
New service RestSecurity; Done.
New service MailServer; Done.
New service dashboardService; Done.
New service iPlanetAMAuthOATHService; Done.
Add Organization schema to sunFAMSAML2Configuration; Done.
Upgrade sunAMAuthHOTPService; Done.
Upgrade sunAMAuthADService; Done.
Upgrade sunAMAuthOAuthService; Done.
Upgrade iPlanetAMAuthCertService; Done.
Upgrade sunIdentityRepositoryService; Done.
Upgrade iPlanetAMPasswordResetService; Done.
Upgrade iPlanetAMSessionService; Done.
Upgrade iPlanetAMAuthService; Done.
Upgrade iPlanetAMAuthLDAPService; Done.
Upgrade sunAMAuthDataStoreService; Done.
Upgrade AgentService; Done.
New sub schema sunIdentityRepositoryService; Done.
New sub schema AgentService; Done.
Delete service sunFAMLibertyInteractionService; Done.
Delete service sunFAMLibertySecurityService; Done.
Creating entitlement application type crestPolicyService; Done.
Creating entitlement application crestPolicyService; Done.
Re-enabling Generic LDAPv3 Data Store; Done.
Upgrading data store embedded; Done.
Updating Platform Properties; Done.
Writing Upgrade Log; Done.

Upgrade Complete.
```

For additional information about the command-line tool, see the reference documentation for upgrade.jar(1) in the *OpenAM Reference*.

8. If you installed OpenAM using an external directory server as the configuration store, add an access control instruction (ACI) to the external directory to give the OpenAM administrative user server-side sorting privileges.

The ACI should be similar to the following:

```
aci: (targetcontrol="1.2.840.113556.1.4.473")(version 3.0;acl "Allow
 server-side sorting"; allow (read)(userdn = "ldap:///
 uid=openam,ou=admins,dc=example,dc=com");)
```

See Section 1.5, "Preparing an External Configuration Data Store" in the *OpenAM Installation Guide* for more information about using an external directory server as the OpenAM configuration store.

9. If you want to configure the upgraded system for the Core Token Service (CTS), read Chapter 6, "Configuring the Core Token Service" in the *OpenAM Installation Guide*. For a list of supported directory services, see the Section 2.4, "Data Store Requirements" in the *OpenAM Release Notes*

10. Referral policies are not supported in OpenAM 13.5. If your OpenAM deployment has referral policies, the following warning message will appear when you upgrade your OpenAM server to OpenAM 13.5:

    Referrals found that require removing

    OpenAM will take the following actions during the upgrade:

    - Removing all referral policies from your OpenAM configuration.
    - Copying resource types and policy sets associated with removed referral policies to the realms targeted by the referral policies.

    For example, suppose you had an OpenAM 12 deployment with a referral policy in realm A, and that referral policy referred to policies in realm B. During an upgrade, OpenAM would delete the referral policy in realm A and copy all the resource types and policy sets associated with the deleted referral policy from realm A to realm B.

    After upgrading to OpenAM 13.5, you are responsible for reconfiguring OpenAM so that policy evaluation that previously depended upon referrals continues to function correctly. You might need to take one or both of the following actions:

    - Reconfiguring your policy agent with the realm and policy set [1] that contain policies to be evaluated when that agent requests a policy decision from OpenAM. Previously, you might have configured the agent to use a realm that contained a referral policy. Because referral policies are not supported in OpenAM 13.5, this is no longer possible.

[1] The agent configuration UI refers to a policy set as an application.

For more information about configuring an agent with a realm and policy set, see Section 4.2, "Working With Realms and Policy Agents" in the *OpenAM Administration Guide*.

- Copying or moving a policy or a group of policies. OpenAM 13.5 has new REST API endpoints that let you copy and move policies. This functionality might be helpful when migrating away from policy deployments that use referral policies. For more information about the REST endpoints that let you copy and move policies, see Section 2.1.2.6.6, "Copying and Moving Policies" in the *OpenAM Developer's Guide*.

11. Validate that the service is performing as expected.

12. Allow client application traffic to flow to the upgraded site.

Procedure 2.2. To Complete Upgrade from OpenAM 11.0.x

After upgrade from OpenAM 11.0.x, all OAuth 2.0 client configurations inherit the default response types:

- `code`
- `token`
- `id_token`
- `code token`
- `token id_token`
- `code id_token`
- `code token id_token`

1. For each OAuth 2.0 client configuration, edit the list of response types to remove any that are not supported or not required.

2. For each OAuth 2.0 client configuration, update the client password.

   As part of a fix for OpenID Connect ID Token signing, the password storage format for OAuth 2.0 clients has changed. OpenAM now stores client passwords using reversible encryption. OpenAM 11.0 stores client passwords using a one-way hash algorithm, and therefore the passwords cannot be recovered.

   You can update the client password by using either OpenAM console or the **ssoadm update-agent** command with the `--attributevalues` option to update the value of the `userpassword` attribute.

Procedure 2.3. To Complete Upgrade from OpenAM 13.0.x

If you configured one or more JDBC audit event handlers in OpenAM 13.0.x, make the following changes to the audit tables' schema:

1. Run the following command on Oracle databases that support OpenAM audit event handlers:

   ```
   ALTER TABLE am_auditaccess ADD (response_detail CLOB NULL);
   ```

   This command adds the `response_detail` column to the `am_auditaccess` table.

2. Run the following commands on MySQL databases that support OpenAM audit event handlers:

   ```
   ALTER TABLE audit.am_auditconfig CHANGE COLUMN configobjectid objectid VARCHAR(255);
   ALTER TABLE audit.am_auditaccess ADD COLUMN response_detail TEXT NULL;
   ```

   The commands change the name of the `configobjectid` column in the `am_auditconfig` table to `objectid` and add the `response_detail` column to the `am_auditaccess` table.

3. If you use databases other than Oracle or MySQL to support OpenAM audit event handlers, review their schema.

   If the `am_auditconfig` table has a column named `configobjectid`, change that column's name to `objectid`.

   If the `am_auditaccess` table does not have a column named `response_detail`, add that column to the table's schema.

Chapter 3

Migrating Legacy Servers

Rather than upgrade legacy servers (running OpenSSO or Sun Access Manager, or an OpenAM version that is no longer supported), you instead manually migrate from your existing deployment to a new deployment.

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

Procedure 3.1. To Upgrade A Legacy Deployment

1. Prepare your customized OpenAM server .war file.
2. Prepare a new deployment, installing servers from the new, customized .war file, starting with the instructions in Chapter 2, "Installing OpenAM Core Services" in the *OpenAM Installation Guide*.
3. After installation, configure the new servers in the same way as the old servers, adapting as necessary.

   You can use the **ssoadm do-batch** command to apply multiple changes with one command.
4. Validate that the new service is performing as expected.
5. Redirect client application traffic from the old deployment to the new deployment.

Chapter 4

Upgrading OpenAM Components

This chapter is concerned with upgrades for policy agents, OpenAM tools, and services.

Procedure 4.1. To Upgrade Web Policy Agents

1. Back up the policy agent installation and configuration directories.

   Also back up the configuration if it is stored centrally in OpenAM.

2. Redirect client traffic away from the protected application.

3. Stop the web server where the policy agent is installed.

4. Remove the old policy agent as described in the *OpenAM Web Policy Agent User's Guide*.

If the uninstallation process has changed, refer to the version of the *Web Policy Agent Installation Guide* that corresponds to your web policy agent.

5. Install the new policy agent using the existing configuration.
6. Start the web server where the policy agent is installed.

   For new features, the policy agent uses the default configuration until you make changes.

7. Validate that the policy agent is performing as expected.
8. Allow client traffic to flow to the protected application.

Procedure 4.2. To Upgrade Java EE Policy Agents

1. Back up the policy agent installation and configuration directories.

   Also back up the configuration if it is stored centrally in OpenAM.

2. Redirect client traffic away from the protected application.
3. Uninstall the old policy agent.
4. Install the new policy agent.

   For new features, the policy agent uses the default configuration until you make changes.

5. Validate that the policy agent is performing as expected.
6. Allow client traffic to flow to the protected application.

Procedure 4.3. To Upgrade OpenAM Tools

Since OpenAM 10.1, the session tools are no longer needed. Upgrading other tools consists of installing new tools and customizing tools scripts as necessary.

1. Install new versions of the tools.
2. Apply any customizations you made to the scripts, referring to the old tools installation as necessary.
3. Once the new tools are working, you can delete the old tools.

Procedure 4.4. To Upgrade to Elliptical Curve Signature Algorithms for Stateless Sessions and OpenID Connect

OpenAM supports Elliptic Curve Digital Signature Algorithms (ECDSA) for stateless sessions and OpenID Connect in OpenAM 13.5 or later.

1. Generate the public and private keys to use with the ECDSA algorithms using the standard curves parameters using **keytool** and configure stateless session to use ECDSA algorithms as shown in Section 9.8.3, "Configuring Elliptic Curve Digital Signature Algorithms" in the *OpenAM Administration Guide*.

2. Manually add the ECDSA algorithms to OpenAM's OAuth2 provider as follows:

    a. On the OpenAM console, navigate to Configure > Global Services > OAuth2 Provider, and scroll down to ID Token Signing Algorithms supported.

    b. In the New Value field, select `ES256`. Repeat the step to add `ES384` and `ES512`, respectively.

    c. For Token Signing Algorithm, select an ECDSA algorithm, such as ES256.

    d. In the Token Signing RSA/ECDSA public/private key pair field, enter the alias of the ECDSA signing key in the keystore.

    e. Click Save.

3. Update the OpenID Connect Client as follows:

    a. On the Top Level Realm, click Agents, and then click OAuth 2.0/OpenID Connect Client.

    b. In the Agent box, select an existing OIDC client.

    c. In the ID Token Signed Response Algorithm field, enter the ECDSA algorithm. For example, `ES256`, `ES384`, or `ES512`.

    d. Click Save.

Procedure 4.5. To Upgrade User Self Services

OpenAM 13.5 has an improved key management system that allows the user self-service feature to successfully operate in a multi-instance server deployment behind a load balancer. This key management system requires a JCEKS keystore that supports asymmetric and symmetric keys.

To help you decide whether to enable a JCEKS keystore after upgrading to OpenAM 13.5, see the following table:

Table 4.1. User Self Service Feature Upgrade

Upgrading from:	Enabling JCEKS required?
Versions prior to OpenAM 13.0	No
OpenAM 13.0 with the REST-based user self-service feature disabled	No
OpenAM 13.0 with the legacy user self-service feature enabled	No
OpenAM 13.0 with the REST-based user self-service feature enabled	*Yes*

The following steps show how to set up the JCEKS keystore for user self-service:

1. In the OpenAM console, navigate to Configure > Server Defaults > Security > Key Store.
2. Change the Keystore File property to `%BASE_DIR%/%SERVER_URI%/keystore.jceks`.
3. Change the Keystore Type property to `JCEKS`.

   These properties can also be modified on a per-server basis as required by navigating to Deployment > Servers > *`Server Name`* > Security > Key Store.

   For more information about inherited properties, see Section 1.5.1, "Configuring Servers" in the *OpenAM Reference*.
4. Restart the OpenAM server.

Index