OpenAM Release Notes

Version 13

David Goldsmith
Gene Hirayama
Chris Lee

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

Abstract

Notes covering OpenAM prerequisites, fixes, known issues. OpenAM provides open source Authentication, Authorization, Entitlement and Federation software.

Table of Contents

Chapter 1

What's New in OpenAM 13

Before you install OpenAM or update your existing OpenAM installation, read these release notes. Then update or install OpenAM.

1.1 Major New Features in OpenAM 13.0.0

OpenAM 13.0.0 is a major release that introduces exciting new features and functional enhancements to OpenAM.

This release introduces the following product enhancements:

1.1.1 Smarter Security Features

- **ForgeRock Authenticator App and Module**

  OpenAM 13 introduces a new authenticator mobile app for iOS and Android, which generates a one-time password (OTP) for strong multi-factor authentication and is consumed by an associated ForgeRock Authenticator (OATH) Authentication module. The mobile app provides easy delivery and secure provisioning via QR codes with recovery codes in the event of lost or stolen devices.

  For more information, see Section 2.4.10, “Hints for the ForgeRock Authenticator (OATH) Authentication Module” in the *OpenAM Administration Guide*.

- **Contextual Authorization**

  OpenAM 13 now supports contextual authorization, which is a powerful way to build context-based intelligence into policies to protect resources at the time of access. Scriptable conditions can examine environmental context and call external REST policy information points (PIPs) to augment the authorization process.

  These scripts can be used to assess risk, calling up stronger authentication mechanisms only when necessary. These custom scripts increase the level of assurance and intelligence that the service provider has, enabling a more informed interaction with the user.

  For more information, see Chapter 6, in the *OpenAM Developer's Guide*.

- **Universal Authorization**

  OpenAM 13 lets administrators define their own resource types using *custom actions*, which can be used to build solution-specific policies. This allows the OpenAM policy engine to now externalize policy in more situations, such as in IoT projects where devices attempt to access resources other than URLs.

  For more information, see Chapter 3, in the *OpenAM Administration Guide*.

- **SAML2 Authentication Module**

  OpenAM 13 now offers a new authentication module based on the SAML 2.0 specification. The SAML2 authentication module allows federation to be incorporated into authentication chains, leveraging federated identities in stronger multi-factor authentication scenarios.

  For more information, see Chapter 12, in the *OpenAM Administration Guide*.

- **Built-In RADIUS Server Support**

  OpenAM 13 now includes a built-in service enabling it to act as a Remote Authentication Dial-In User Service (RADIUS) server. Administrators can now have a single authentication service for both VPNs that use RADIUS and for access to other protected services.

  For more information, see Chapter 19, in the *OpenAM Administration Guide*.

1.1.2 IoT Features

- **OAuth 2.0 Device Flow**

  OpenAM 13 now supports the de facto OAuth 2.0 device flow for devices to pair devices with minimal or no input capability, such as set-top boxes or Internet of

Things (IoT) devices, with user accounts. This feature offers a standards-based approach to connecting devices and things to services.

For more information, see Section 13.1.1.5, “OAuth 2.0 Device Flow” in the *OpenAM Administration Guide*.

1.1.3 Privacy and Consent Features

- **UMA Authorization Server**

  OpenAM 13 can now act as a fully spec-compliant User-Managed Access (UMA) authorization server, allowing users to control who can gain access to their resources as well as providing a means to manage their own consents.

  OpenAM supports the following UMA features:

  - Resource set registration
  - Resource sharing
  - Resource labelling
  - Pending requests
  - Audit history

  For more information, see the Chapter 15, in the *OpenAM Administration Guide*.

1.1.4 OAuth 2.0/OpenID Connect 1.0 Enhancements

- **OAuth 2.0 Provider Wizards**

  OpenAM 13 now supports OAuth 2.0 provider wizards that ensure optimal configuration for each provider type. These wizards speed up the time to deploy to production environments. The following providers are available:

  - OAuth 2.0 provider
  - OpenID Connect 1.0 provider
  - Mobile Connect provider
  - UMA provider

- **OpenID Certified**

OpenAM 13 is fully conformant with the OpenID Foundation's conformance tests, which rigorously enforce its standards across mobile and web application. Conformance with the standard ensures that customers are not locked in to vendor proprietary solutions.

Figure 1.1. OpenID Certified mark

- **OpenID Connect 1.0 Claims Scripts**

  OpenAM 13 now issues ID tokens that can be augmented with additional claims by means of OIDC claims scripts. This feature makes it easier to build solutions requiring additional identity information.

  For more information, see Section 6.3, “Using the Default Scripts” in the *OpenAM Developer's Guide*.

- **Base URL Source Service**. OpenAM supports a provider service that allows a realm to have a configured option for obtaining the base URL (including protocol) for components that need to return a URL to the client. This service is used to provide the URL base that is used in the `.well-known` endpoints used in OpenID Connect 1.0 and UMA.

  For more information, see Section 14.4, “Configuring the Base URL Source Service” in the *OpenAM Administration Guide*.

1.1.5 CTS Performance Enhancements

- **CTS Performance**

  OpenAM 13 has optimized indexes to improve the performance of the Core Token Service.

1.1.6 Elasticity and Scaling Features

- **Stateless Sessions**

  OpenAM 13 supports two types of sessions: *stateful* and *stateless*.

  Stateful sessions are sessions that reside in the OpenAM server's memory and, if session failover is enabled, are also persisted in the Core Token Service's token store. Stateful sessions have been the only session type supported in previous OpenAM releases.

OpenAM 13 also supports a new type of session: the stateless session. Stateless sessions are sessions in which state information is encoded in OpenAM and sent to clients, but the information from the sessions is not retained in OpenAM's memory. For browser-based clients, OpenAM sets a cookie in the browser that contains the session state. When the browser transmits the cookie back to OpenAM, OpenAM decodes the session state from the cookie.

Stateless sessions can be used for deployments when elasticity is required, for example, cloud deployments in which the server load varies. You can add and remove OpenAM servers to and from a site and the stateless session load should balance horizontally.

For more information, see Chapter 9, in the *OpenAM Administration Guide*.

- **Dynamic Configuration**

  OpenAM 13's many services that previously required a server restart are now hot-swappable.

1.1.7 User Experience Enhancements

- **New Themeable User Interface**

  OpenAM 13 now provides new responsive and rich JavaScript-based user interface themes, providing easier customization.

  For more information, see Section 5.1, “Customizing the End User Interface” in the *OpenAM Installation Guide*.

- **Recording Troubleshooting Information**

  The new **ssoadm start-recording** command lets you initiate events that monitor OpenAM, while saving output that is useful when performing troubleshooting. You can also start a recording event from the `/json/records` endpoint.

  After starting a recording event, you can use the new **ssoadm get-recording-status** command to get the status of the recording event and the new **ssoadm stop-recording** command to terminate the recording event.

  For more information, see Section 24.5, “Recording Troubleshooting Information” in the *OpenAM Administration Guide* and Section 2.1.7, “RESTful Troubleshooting Information Recording” in the *OpenAM Developer's Guide*.

1.1.8 Platform Features

- **Common Self-Service**

OpenAM 13 introduces a new user self-service feature that allows users to register to your web site and reset forgotten passwords. This feature decreases help desk costs as users can on-board and maintain their own accounts. The service is exposed over REST endpoints enabling custom or mobile front-ends to utilize it. The user self-service feature delivers a consistent user experience across the ForgeRock platform (OpenAM, OpenIDM, OpenDJ).

For more information, see Chapter 8, in the *OpenAM Administration Guide*.

- **Common Audit Logging**

  OpenAM 13 introduces the new ForgeRock Common Audit Framework, allowing OpenAM to log all user and administrative activity in a consistent format across the ForgeRock platform. Logs can be written to file, database, or syslog. Common Audit Logging gives administrators a common and consistent audit trail of all user activity across the ForgeRock platform.

  For more information, see Chapter 6, in the *OpenAM Administration Guide*.

- **OpenIG as a Replacement for DAS**

  ForgeRock's OpenIG can act as an intelligent reverse proxy server between clients and the OpenAM Service. When deployed within a DMZ, OpenIG can inspect all traffic and properly forwarding requests to OpenAM.

- **OpenDJ 3.0**

  OpenAM 13 has upgraded its embedded directory service to use the new OpenDJ 3.0 server as its configuration, token, and UMA store.

1.1.9 Developer-friendly Features

- **Scripting Service**

  OpenAM 13 has enhanced its Scripting Service, providing a library and editor that builds authentication scripts (client and server), authorization policy condition scripts, and OpenID Connect Claims gathering scripts. The OpenAM Scripting Service allows easy and fast customization of authentication and authorization services.

  For more information, see Chapter 22, in the *OpenAM Administration Guide*.

- **SOAP STS**

  OpenAM 13 has enhanced its STS solution, adding a SOAP STS solution to the REST STS service in OpenAM 12.0. The SOAP STS service is a token transformation service that allows a mobile app developer who possesses

an OIDC token to generate a SAML assertion to access resources held by a federated service provider.

The SOAP STS is deployed remotely from OpenAM in the following containers:

- Apache Tomcat, versions 6, 7, or 8
- Jetty, versions 7, 8, or 9

1.2 Improvements in OpenAM 13.0.0

The following improvements and additional features were added in this release:

- OPENAM-7235: Fetch additional SSOToken attributes like legacy REST interface
- OPENAM-7123: Allow country-specific localization in XUI
- OPENAM-7109: Allow user to adjust the size of Metadata that can be uploaded by the Common Task "Create SAMLv2 Providers" buttons.
- OPENAM-7055: Improved logic for POST binding Assertion/Response signature check
- OPENAM-6892: Create a Shared Secret Provider plugin for the standard OATH module
- OPENAM-6872: Improved error message in OpenSSOConfigurator
- OPENAM-6534: Update OAuth2 tokeninfo endpoint to be realm-independent
- OPENAM-6272: Allow storing the shared secret in an encrypted format for the OATH Module
- OPENAM-6266: Allow the confirmation email URL in the Forgotten password service to be a relative path
- OPENAM-6236: Add token life time options per OAuth2 client
- OPENAM-6069: Make org.forgerock.openam.agents.config.policy.evaluation.* optional
- OPENAM-5859: Support the form_post OAuth 2.0 response mode
- OPENAM-5785: Allow ssoadm to import and export agent configurations with hashed passwords
- OPENAM-5759: Update OAuth2 to display the token and user information in the OAuth2Provider.access log

- OPENAM-5755: Prevent duplicate metaAliases in SAML2 entities
- OPENAM-5695: Allow admin users to update user's password without the old password
- OPENAM-5477: Add configuration to allow OAuth2 Refresh Tokens to never expire
- OPENAM-5419: Update TokenExpired exception message to be consistent
- OPENAM-5332: Update OAuth2 RefreshTokenServerResource to check the clientID case insensitively
- OPENAM-5311: Update Default timelimit in Netscape SDK to be configurable
- OPENAM-5260: Provide option to only sign Response when using HTTP-POST binding
- OPENAM-5097: Update LDAP and AD auth modules to support startTLS extended operation
- OPENAM-5065: PLLClient should call getErrorStream() to get response body on IOException.
- OPENAM-4923: Update Windows Desktop SSO module to allow whitelisting Kerberos realms/KDCs
- OPENAM-4459: OpenID Connect attribute mappings should be localizable
- OPENAM-4177: Update OAuth2 access_token endpoint to handle auth chain with non-name/password callback
- OPENAM-4103: Allow sending AuthnRequests without the RequestedAuthnContext element
- OPENAM-3743: Update OAuth2 authorization consent pages to be fully localizable
- OPENAM-3714: Update DJLDAPv3Repo to support StartTLS
- OPENAM-3493: Update SAML to support multiple keys (key rollover)
- OPENAM-2238: Support extensibility of auth context classes as described in the SAMLv2 spec
- OPENAM-1900: Provide support for more XML signatures types in SAML query string verification process
- OPENAM-1650: Implement REST services for manipulating Session properties

- OPENAM-1631: Add option to enable debug logging of decrypted SAML assertions

1.3 Security Advisories

ForgeRock issues security advisories in collaboration with our customers and the open source community to address any security vulnerabilities transparently and rapidly. ForgeRock's security advisory policy governs the process on how security issues are submitted, received, and evaluated as well as the timeline for the issuance of security advisories and patches.

For more information on ForgeRock's security advisory policy, click the following link: http://www.forgerock.com/services/security-policy/

The following security advisory concerns vulnerabilities that have been fixed in this release of OpenAM:

- OpenAM Security Advisory #201601

Chapter 2

Before You Install OpenAM Software

This chapter covers software and hardware prerequisites for installing and running OpenAM server software.

ForgeRock supports customers using the versions specified here. Other versions and alternative environments might work as well. When opening a support ticket for an issue, however, make sure you can also reproduce the problem on a combination covered here.

2.1 OpenAM Operating System Requirements

ForgeRock supports customers using OpenAM server software on the following operating system versions:

Table 2.1. Supported Operating Systems

Operating System	Version
Red Hat Enterprise Linux, Centos	6, 7
SuSE	11
Ubuntu	12.04 LTS, 14.04 LTS
Solaris x64	10, 11
Solaris Sparc	10, 11

Operating System	Version
Windows Server	2008, 2008 R2, 2012, 2012 R2

2.2 Java Requirements

Table 2.2. JDK Requirements

Vendor	Version
Oracle JDK	7, 8
IBM SDK, Java Technology Edition (Websphere only)	7

2.3 OpenAM Web Application Container Requirements

Table 2.3. Web Containers

Web Container	Version
Apache Tomcat	7, 8
Oracle WebLogic Server	12c
JBoss Enterprise Application Platform	6.1+
JBoss Application Server	7.2+
WildFly AS	9
IBM WebSphere	8.0, 8.5

The web application container must be able to write to its own home directory, where OpenAM stores configuration files.

2.4 Data Store Requirements

Table 2.4. Supported Data Stores

Data Store	Version	CTS Datastore	Config Datastore	User Datastore	UMA Datastore
Embedded OpenDJ	3.0	✓	✓	✓	✓
External OpenDJ	2.6, 2.6.2, 3.0	✓	✓	✓	✓
Oracle Unified Directory	11g			✓	
Oracle Directory Server Enterprise Edition	11g			✓	
Microsoft Active Directory	2008, 2008 R2, 2012, 2012 R2			✓	
IBM Tivoli Directory Server	6.3			✓	

2.5 Supported Clients

The following table summarizes supported clients:

Table 2.5. Supported Clients

Client Platform	Native Apps[a]	Chrome 16+[b]	IE 9+	Firefox 3.6+	Safari 5+
Windows 7 or later	✓	✓	✓	✓	✓
Mac OS X 10.8 or later	✓	✓		✓	
Ubuntu 13.04 LTS or later	✓	✓		✓	✓
iOS 7 or later	✓	✓			✓
Android 4.3 or later	✓	✓			

[a] *Native Apps* is a placeholder to indicate OpenAM is not just a browser-based technology product. An example of a native app would be something written to use our REST APIs, such as the sample OAuth 2.0 Token Demo app.
[b] Chrome, Firefox, and Safari are configured to update automatically, so customers will typically running the latest versions. The versions listed in the table are the minimum required versions.

2.6 Supported Upgrade Paths

ForgeRock supports upgrades from the following versions to this version of OpenAM:

Table 2.6. Supported Upgrades

Version	Upgrade Supported?
OpenAM 9.0.x	No
OpenAM 9.5.x	No
OpenAM 10.0.x	Yes
OpenAM 11.0.x	Yes
OpenAM 12.0.x	Yes

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

2.7 Special Requests

If you have a special request regarding support for a combination not listed here, contact ForgeRock at info@forgerock.com.

Chapter 3

Installing or Upgrading

This chapter covers installing and upgrading OpenAM 13 software.

Before you install OpenAM or upgrade your existing OpenAM installation, read these release notes. Then, install or upgrade OpenAM.

- If you are installing OpenAM for the first time, see the OpenAM Installation Guide.
- If you have already installed OpenAM, see the OpenAM Upgrade Guide.

  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.

Chapter 4

Changes and Deprecated Functionality

This chapter covers both major changes to existing functionality, and also deprecated and removed functionality.

4.1 Important Changes to Existing Functionality

These changes are new in OpenAM 13:

- **New Attribute Required in Authentication Service Definition**. OpenAM 13 requires that schemas in the definition of an authentication service contain `resourceName` attributes.

  The attributes are not added to custom authentication service definitions when upgrading from a previous version, so must be added manually.

  The specific changes required in the service definition schema are:

  - The `Schema` element in the service definition must contain a `resourceName` attribute. This value is used to refer to the service when managing the service using REST.

    For example:

```
<Schema
 serviceHierarchy="/DSAMEConfig/authentication/iPlanetAMAuthSampleAuthService"
 i18nFileName="amAuthSampleAuth"
 revisionNumber="10"
 i18nKey="sampleauth-service-description"
 resourceName="mySampleAuthService">
```

- Any `SubSchema` elements in the service definition must contain a `resourceName` attribute, with a value of `USE-PARENT`.

  For example:

```
<SubSchema
 name="serverconfig"
 inheritance="multiple"
 resourceName="USE-PARENT">
```

An example of a service definition compatible with OpenAM 13 can be found in Section 4.3.6, “The Sample Auth Service Configuration” in the *OpenAM Developer's Guide*.

Procedure 4.1. To Add Required Attributes to Custom Service Definition Schemas

You can add the required attributes either before or after upgrading to OpenAM 13. The steps in this procedure cover adding the attributes before upgrading.

1. If you have not already done so, install and configure a tool for altering the contents of the OpenDJ configuration store, for example the OpenDJ Control Panel or Apache Directory Studio.

2. Connect to the embedded configuration store using the same bind DN credentials as configured in OpenAM. The default is `cn=Directory Manager`.

3. In the directory tree of the configuration store, locate the `sunServiceSchema` attribute for your custom service definition under `ou=services`.

   For example, on a default install the definition for the data store service is located here: `ou=1.0,ou=sunAMAuthDataStoreService,ou=services,dc=openam,dc=forgerock,dc=org`

4. Edit the XML stored within the `sunServiceSchema` attribute, adding the required `resourceName` attribute to `Schema` and `SubSchema` elements.

5. Commit the changes to the configuration store, and proceed to upgrade OpenAM.

Failure to add the required attributes will result in the OpenAM 13 user interface being unable to view or edit custom services, or create or edit

authentication modules based on them after upgrade. You may also see a `Not found error` message displayed in the administration console when creating or editing authentication modules.

- **Changes to SAML 2.0 NameID Persistence**. OpenAM's SAML 2.0 account management and NameID persistence logic has been updated to work better with non-persistent NameID formats.

  The NameID persistence logic is summarized as follows:

  ```
  Persistent NameID         -> NameID MUST be stored
  Transient NameID          -> NameID MUST NOT be stored
  Ignored user profile mode -> NameID CANNOT be stored (fails if used in
                               combination with persistent NameID-Format)
  For any other case        -> NameID MAY be stored based on customizable logic
  ```

  The following changes have been made on the identity provider side:

  - **New Setting: idpDisableNameIDPersistence**. OpenAM now provides a new setting, `idpDisableNameIDPersistence`, which disables the storage of the NameID values for all NameIDs issued by that IdP instance, as long as the NameID-Format is anything but `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`.

  - **SP's spDoNotWriteFederationInfo Repurposed**. The SP's `spDoNotWriteFederationInfo` setting has been repurposed. It no longer is specific to unspecified NameID-Formats. Now, it affects all non-persistent NameID-Formats, similar to the `idpDisableNameIDPersistence` setting in the IdP configuration.

  - **NameID Lookup Changes**. The NameID lookup mechanism has been modified, so that it only tries to look up existing NameID values for the user if the NameID is actually persisted for the corresponding NameID-Format.

  - **New Method in the IDPAccountMapper Interface**. The `IDPAccountMapper` interface has been extended with the following new method:

```
/**
* Tells whether the provided NameID-Format should be persisted in the user data
*       store or not.
*
* @param realm The hosted IdP's realm.
* @param hostEntityID The hosted IdP's entityID.
* @param remoteEntityID The remote SP's entityID.
* @param nameIDFormat The non-transient, non-persistent NameID-Format in question.
* @return true if the provided NameID-Format should be persisted
*       in the user data store, false otherwise.
*/
public boolean shouldPersistNameIDFormat(String realm, String hostEntityID,
       String remoteEntityID,String nameIDFormat);
```

The default implementation of `shouldPersistNameIDFormat` in `DefaultIDPAccountMapper` first checks whether `idpDisableNameIDPersistence` is enabled in the hosted IdP configuration. If `idpDisableNameIDPersistence` is disabled, the logic advances and accesses the remote SP's `spDoNotWriteFederationInfo` flag.

The following changes have been made on the service provider side:

- **Changes to SPAccountMapper**. The `SPAccountMapper` implementations now no longer need to perform reverse lookups using the received NameID value. The `SPACSUtils` now performs the reverse lookup if the NameID-Format should be persisted. This change was made to ensure that NameID values are only persisted in the data store if they have not been stored there previously.

- **SP's spDoNotWriteFederationInfo Repurposed**. The SP's `spDoNotWriteFederationInfo` setting has been repurposed. It no longer is specific to unspecified NameID-Formats. It affects all non-persistent NameID-Formats.

- **New Method in the SPAccountMapper Interface**. The `SPAccountMapper` interface has been extended with the following new method:

```
/**
* Tells whether the provided NameID-Format should be persisted in the user data
*       store or not.
*
* @param realm The hosted SP's realm.
* @param hostEntityID The hosted SP's entityID.
* @param remoteEntityID The remote IdP's entityID.
* @param nameIDFormat The non-transient, non-persistent NameID-Format in question.
* @return true if the provided NameID-Format should be persisted
*       in the user data store, false otherwise.
*/
public boolean shouldPersistNameIDFormat(String realm, String hostEntityID,
       String remoteEntityID, String nameIDFormat);
```

The default implementation of `shouldPersistNameIDFormat` in `DefaultLibrarySPAccountMapper` checks whether `spDoNotWriteFederationInfo` is enabled in the hosted SP configuration.

For more information, see OPENAM-3470.

- **AD/LDAP/RADIUS Authentication Modules Allow More Than One Primary/ Secondary Server**. The Active Directory, LDAP, and RADIUS authentication modules now allow one or more servers to be designated as primary or secondary servers.

  When authenticating users from a directory server that is remote to OpenAM, set the primary server values, and optionally, the secondary server values. Primary servers have priority over secondary servers.

  ssoadm attributes are: primary is `iplanet-am-auth-ldap-server`; secondary is `iplanet-am-auth-ldap-server2`.

  Both properties take more than one value; thus, allowing more than one primary or secondary remote server, respectively. Assuming a multi-data center environment, OpenAM determines priority within the primary and secondary remote servers, respectively, as follows:

  - Every LDAP server that is mapped to the current OpenAM instance has highest priority.

    For example, if you are connected to `openam1.example.com` and `ldap1.example.com` is mapped to that OpenAM instance, then OpenAM uses `ldap1.example.com`.

  - Every LDAP server that was not specifically mapped to a given OpenAM instance has the next highest priority.

    For example, if you have another LDAP server, `ldap2.example.com`, that is not connected to a specific OpenAM server and if `ldap1.example.com` is unavailable, OpenAM connects to the next highest priority LDAP server, `ldap2.example.com`.

  - LDAP servers that are mapped to different OpenAM instances have the lowest priority.

    For example, if `ldap3.example.com` is connected to `openam3.example.com` and `ldap1.example.com` and `ldap2.example.com` are unavailable, then `openam1.example.com` connects to `ldap3.example.com`.

  For more information, see OPENAM-3575.

- **New XUI Reverse Proxy Support Option**.

A new option for controlling caching in the XUI when behind a reverse proxy is available in this release. The option is disabled by default when upgrading to preserve previous behavior, and enabled in clean installs.

If reverse proxy support in the XUI is required after an upgrade from 12.0.0, delay enabling the `XUI Reverse Proxy Support` option long enough that cached JavaScript files on end-user clients have expired, for example 30 days. Failure to do so may result in users being redirected to `http://null:8080`.

- **Legacy User Self Service Endpoints Disabled by Default**.

  The REST endpoints used by the legacy user self service features, such as registering for an account or resetting a forgotten password, are now disabled by default.

  Legacy deployments should migrate to the new user self-service features in OpenAM 13, see Chapter 8, in the *OpenAM Administration Guide*.

  To restore the legacy endpoints, enable the Configuration > Global > Legacy User Self Service > Legacy Self-Service REST Endpoint option.

  

  Warning

  Restoring the legacy self service endpoints allows REST requests crafted such that the body of the self-service email contains a malicious URL that end users may visit, hiding the correct OpenAM URL that is appended to the end of the email body.

- **Destination After Successful Self-Registration Option Removed**.

  The new user self-service workflow will always display the success page after completing self-registration. The option to choose the behavior has been removed.

- **REST Endpoint Changes**

  Version 3.0 of the `/users` endpoint is provided in this release of OpenAM. The response differs from version 2.0 of the endpoint, which remains available for backwards compatibility.

  The new version of the endpoint returns details about all users. The previous version only returned a list of usernames.

Version 3.0 of the `/users` endpoint does not support the following `_action` values:

```
https://openam.example.com:8443/openam/json/users/?_action=register
https://openam.example.com:8443/openam/json/users/?_action=confirm
https://openam.example.com:8443/openam/json/users/?_action=anonymousCreate
https://openam.example.com:8443/openam/json/users/?_action=forgotPassword
https://openam.example.com:8443/openam/json/users/?_action=forgotPasswordReset
```

Responses to Different Versions of the `/users` Endpoint

In this section, long URLs are wrapped to fit the printed page, and some of the output is formatted or truncated for easier reading.

Version 3.0 of the `/users` endpoint:

Version 2.0 of the `/users` endpoint:

```
$ curl \
--header "iPlanetDirectoryPro: AQIC5w...2NzEz*" \
--header Accept-API-Version: protocol=1.0,resource=2.0 \
"https://openam.example.com:8443/openam/json/users?_queryId=*"
{
    "result": [
        "amAdmin",
        "demo"
    ],
    "resultCount": 2,
    "pagedResultsCookie": null,
    "totalPagedResultsPolicy": "NONE",
    "totalPagedResults": -1,
    "remainingPagedResults": -1
}
```

- **Workaround for java.lang.VerifyError in WebSphere**.

  When loading classes from OpenAM within WebSphere Application Server using the IBM Technology for JVM and Apache Axis2 framework, a `java.lang.VerifyError JVMVRFY013 class loading constraint violated` error may occur. For more information on the `java.lang.VerifyError` error, see "java.lang.VerifyError: JVMVRFY013 class loading constraint violated" Error.

  Procedure 4.2. Fixing a WebSphere java.lang.VerifyError Error

  1. Remove the following JARs from the `WEB-INF/lib` directory in the `openam.war` file:

     - `jaxp-api-1.4.2.jar`
     - `xercesImpl-2.11.0.jar`
     - `xml-apis-2.11.0.jar`
     - `xml-resolver-2.11.0.jar`
     - `xml-serializer-2.11.0.jar`

     For instructions on how to expand the `openam.war` file, make changes to `bootstrap.properties` file, and then rebuild the `openam.war` file, see Procedure 1.6, “To Prepare OpenAM for JBoss and WildFly” in the *OpenAM Installation Guide*.

  2. Set the following custom JVM properties on the WebSphere server:

```
-Djavax.xml.soap.MessageFactory=com.sun.xml.internal.messaging.saaj.soap.ver1_1.SOAPMessageFactory1
-Djavax.xml.soap.SOAPFactory=com.sun.xml.internal.messaging.saaj.soap.ver1_1.SOAPFactory1_1Impl
-Djavax.xml.soap.SOAPConnectionFactory=com.sun.xml.internal.messaging.saaj.client.p2p.HttpSOAPConne
-Djavax.xml.soap.MetaFactory=com.sun.xml.internal.messaging.saaj.soap.SAAJMetaFactoryImpl
-Dcom.ibm.websphere.webservices.DisableIBMJAXWSEngine=true
```

3. Restart the WebSphere server.

- **Different return type for GetUserInfo method of ScopeValidator interface**.

  The return type for the `getUserInfo` method of the `org.forgerock.oauth2.core.ScopeValidator` interface, formerly `Map<String, Object>`, is now `org.forgerock.oauth2.core.UserInfoClaims`. The new return type lets callers of the `getUserInfo` method see values of users' claims.

  This change affects OAuth 2.0 scope validator plugins. For more information, see Section 4.2, “Customizing OAuth 2.0 Scope Handling” in the *OpenAM Developer's Guide*.

- **Oracle Directory Server Enterprise Edition no longer supported for the OpenAM configuration store**.

  In previous versions, it was possible to deploy the OpenAM configuration store in an external Oracle Directory Server Enterprise Edition instance.

  In OpenAM 13, this is no longer possible. You must deploy the OpenAM configuration store in an OpenDJ server instance: either the embedded OpenDJ directory server instance that is installed together with OpenAM, or in an external server instance.

4.2 Deprecated Functionality

The following features are deprecated in OpenAM 13:

- The OpenAM Logging, User Self Service, and Password Reset Services are deprecated. The User Self Service has been renamed to Legacy User Self Service. New audit logging and user self-service capabilities are available in OpenAM 13.0.0. See Section 1.1.8, "Platform Features" for more information.

- The classic JATO-based UI is deprecated for the end-user pages and replaced in OpenAM with the JavaScript-based XUI as a replacement. The classic UI for end user pages is likely to be removed in a future release.

- Listing tokens with the `/frrest/oauth2/token/?_queryId` method is deprecated. Improved `_queryFilter` support will be added to replace the `_queryId` method.

- The Device Print Service is deprecated. For information on replacement device identification features, see Section 2.4.7, “Hints for the Device ID (Match) Authentication Module” in the *OpenAM Administration Guide*.

4.3 Removed Functionality

The following functionality has been removed from OpenAM:

- The `/identity` endpoints that were not previously deprecated are no longer in OpenAM:
  - `/log`
  - `/getCookieNamesForToken`
  - `/getCookieNamesToForward`
- Use of the legacy Netscape LDAP SDK is replaced by the OpenDJ SDK.
- The `sun-idrepo-ldapv3-config-connection-mode` property replaces `sun-idrepo-ldapv3-config-ssl-enabled`, which has been removed from the configuration schema (`sunIdentityRepositoryService`).

  For more information, see OPENAM-3714.
- The `openam-auth-ldap-connection-mode` property replaces `iplanet-am-auth-ldap-ssl-enabled`, which has been removed from the configuration schema (`sunAMAuthADService` and `iPlanetAMAuthLDAPService`).

  For more information, see OPENAM-5097.
- **New `openam.deserialisation.classes.whitelist` Property**. OpenAM uses the JATO framework for some console pages and for legacy login pages. The JATO framework uses serialized Java objects to maintain state during the console session.

  To ensure that the serialized objects have not been exploited by a malicious user, OpenAM now provides a new `openam.deserialisation.classes.whitelist` property that lists valid classes when OpenAM performs object deserialization. The default should work for most deployments.

  You can access and update the property on the OpenAM console by navigating to Configuration > Servers and Sites > Default Server Settings > Security > Object Deserialisation Class Whitelist.

  For more information, see OPENAM-5925.

- REST services relying on the following endpoints have been removed from OpenAM.
  - `/identity/attributes`
  - `/identity/authenticate`
  - `/identity/authorize`
  - `/identity/create`
  - `/identity/delete`
  - `/identity/isTokenValid`
  - `/identity/logout`
  - `/identity/read`
  - `/identity/search`
  - `/identity/update`
  - `/json/[`*`realm/`*`]referrals`
  - `/ws/1/entitlement/decision`
  - `/ws/1/entitlement/decisions`
  - `/ws/1/entitlement/entitlement`
  - `/ws/1/entitlement/entitlements`
  - `/ws/1/entitlement/listener`
  - `/ws/1/entitlement/privilege`
  - `/ws/1/token`
- The Persistent Cookie (Legacy) settings in the Core Authentication module have been removed, along with the following properties:
  - `com.iplanet.am.cookie.timeToLive`
  - `openam.session.allow_persist_am_cookie`
  - `openam.session.persist_am_cookie`

For information on how to configure persistent cookies in this release, see Section 2.4.19, “Hints for the Persistent Cookie Module” in the *OpenAM Administration Guide*.

- The server-only WAR file has been removed from the OpenAM distribution.
- The Distributed Authentication Service (DAS) has been removed.
- Referral policies are no longer available in OpenAM. If you are upgrading from a previous version of OpenAM and currently use referral policies, please refer to the OpenAM Upgrade Guide for migration information.
- Specifying a realm in POST data is no longer supported. A number of other methods are supported, such as specifying the realm as a query parameter.

Chapter 5

Fixes, Limitations, and Known Issues

OpenAM issues are tracked at https://bugster.forgerock.org/jira/browse/OPENAM. This chapter covers the status of key issues and limitations at release 13.

5.1 Key Fixes

The following bugs were fixed in release 13. For details, see the OpenAM issue tracker.

5.1.1 Key Fixes in OpenAM 13.0.0

The following important issues were fixed in this release:

- OPENAM-7382: HTTP GET to uma/auditHistory with param "sortKeys=-eventTime" returned 500 server error
- OPENAM-7334: Client Authentication method not compliant with OpenID standard
- OPENAM-7021: XUI login script queries "/openam/json/users?realm=/?_action=idFromSession"
- OPENAM-6977: Validate OIDC script returns "No privilege mapping for requested action validate"

- OPENAM-6976: OAuth2 Error Page on oauth2/authorize with valid params and cookie
- OPENAM-6867: changePassword REST endpoint is not returning LDAP issues that are related to a user mistake.
- OPENAM-6613: Updating Hosted IDP Authentication Context Mapper does not save values
- OPENAM-6553: Fix Social Authentication in subrealms
- OPENAM-6545: ServerInfoResource should attempt to cache ServiceConfigs per realm rather than creating one on each request
- OPENAM-6499: Configuration store servers are not listed in Directory Configuration
- OPENAM-6468: InvalidClassException with certauth after #201505-01 patch
- OPENAM-6457: DirectoryContentUpgrader causes Entry Already Exists exception for CTS suffix when upgrading OpenAM
- OPENAM-6385: Revoking access to individual resource using XUI fails
- OPENAM-6384: XUI: Sharing resource twice (with another user) fails
- OPENAM-6377: CTSOperations is currently performing setLatestAccessTime on a local token, rather than the remote one.
- OPENAM-6374: Registering UMA resource sometimes gives error
- OPENAM-6293: XUI freezes at startup when serverinfo service call fails
- OPENAM-6156: orderedlist uitype in service config breaks when updated
- OPENAM-6056: LoginViewBean does not correctly handle empty ChoiceCallbacks
- OPENAM-6039: Asynchronous queue for OAuth2 Tokens can result in token validation failures
- OPENAM-6000: Accessing XUI through a FQDN that is resolvable but not mapped throws an internal server error
- OPENAM-5894: Can't update WindowsDesktopSSO module with ssoadm
- OPENAM-5841: Realm override query parameter on login not overriding realm
- OPENAM-5826: Zero Page Login disallowed after OPENAM-sec-201503 CAS is applied

- OPENAM-5804: Forgot password in XUI with a sub-realm when using RFC3986 specs not redirecting correctly
- OPENAM-5721: WindowsDesktopSSO trusted realm list doesn't work
- OPENAM-5690: Get an Access Token From SAML 2.0 on 12.0.0 uses grant type saml2-bearer, but TokenEndpoint is not defined in OAuth2Application
- OPENAM-5660: NPE when the keyalias does not exist or does not contain a certificate
- OPENAM-5623: CTS uses inefficient search for coreTokenId=
- OPENAM-5598: Adaptive Risk auth module can not be used in auth chain if the username in sharedstate map does not 'match' the search attribute of the data store
- OPENAM-5508: REST with Realm/DNS Aliases causes unexpected results
- OPENAM-5488: Upgrade fails from OpenAM 11 to OpenAM 12 with NPE from OAuth2 client profile
- OPENAM-5472: NPE in #setAttributes when IdRepo fails to read directory schema
- OPENAM-5460: AD auth module does not provide property 'iplanet-am-auth-ldap-ssl-trust-all' as LDAP auth module does
- OPENAM-5451: Resource based authentication does not work as expected in 12 (with legacy UI)
- OPENAM-5421: TokenResource ignores query string passed from client
- OPENAM-5417: Policy Conditions of same type can not be combined in OpenAM 12
- OPENAM-5411: OpenAM is filling the ResponseLocation with a null instead of an empty string
- OPENAM-5396: Malformed exp parameter in ID token
- OPENAM-5386: Policy editor doesn't always use realm-specific REST endpoints
- OPENAM-5383: CTS Reaper fails if simple paged control is not present in response
- OPENAM-5381: Specifying an external user store when using configurator tool is not being processed correctly

- OPENAM-5321: Cross realm session upgrade not handled properly by XUI
- OPENAM-5317: 1st. character from realm value is deleted from endpoint /json/authenticate?realm=myRealm"
- OPENAM-5312: Initialization of a ServiceSchemaManager may block retrieval of already cached instances
- OPENAM-5304: XUI does not work behind HTTP reverse proxy if HTTP host header is not preserved
- OPENAM-5252: DJLDAPv3Repo returns different error code when DN cache is enabled
- OPENAM-5237: OAuth2 authorization consent page uses absolute URL in FORM tag
- OPENAM-5208: SAML2 SLO error on IDP with Session Synchronization when SP does not support SOAP binding
- OPENAM-5197: OAuth2 client fails to add access_token to tokeninfo call
- OPENAM-5183: CTS port settings are reverted to default when doing upgrade from AM 11 to AM 12
- OPENAM-5148: URL links in email sent from REST forgotPassword or register is not URLEncoded
- OPENAM-5120: SAML2 SP in a sub-realm not fully functional after OPENAM-474
- OPENAM-5082: DJLDAPv3Repo setAttributes may add unnecessary objectclasses to modifyRequest.
- OPENAM-5034: Legacy password pages unable to handle special characters in username
- OPENAM-4919: DNMapper.realmNameToAMSDKName logic adding extra = when checking against orgAttr
- OPENAM-4856: HOTP auth module can not be used in auth chain if the username in sharedstate map does not 'match' the search attribute of the data store
- OPENAM-4804: SAE fails with No_App_Attrs:https error
- OPENAM-4644: Log file rotation isn't respected
- OPENAM-4614: MergeAll Option cause a desynchronisation of the log rotation

- OPENAM-4605: Unable to install OpenAM Configuration Data Store into an 'ou' through the console
- OPENAM-4413: Agent sessions are affected by active session quotas when com.iplanet.am.session.agentSessionIdleTime is used
- OPENAM-4344: OAuth2 SAML bearer grant does not work
- OPENAM-4333: OAuth2 endpoint doesn't honour realm DNS aliases - must specify realm via URL query string
- OPENAM-4195: SAML2token saved in CTS with hex tokenId but read without converting to hex
- OPENAM-4164: AgentsRepo may have cached stale ServiceConfigImpl, returning incorrect agent profile data.
- OPENAM-3877: Changing password through new REST endpoint fails if default AuthN chain needs more than just the password to authenticate
- OPENAM-3856: AMAuthenticationManager can get incorrectly initialized for subrealms
- OPENAM-3575: LDAP auth module fails if more than one LDAP server is configured as primary/secondary LDAP server
- OPENAM-3296: ssoadm uses LDAP auth module first to authenticate amadmin
- OPENAM-2348: set-realm-svc-attrs: "Not a supported type: realm"
- OPENAM-2137: DSConfigMgr can hide exception root causes
- OPENAM-816: ssoadm authentication depends on the sunEnableModuleBasedAuth=true
- OPENAM-718: Agent group membership lost after backup/restore
- OPENAM-273: com.sun.identity.policy.PolicyManager, when used in client API, does not work across multiple SSO sessions in a single JVM instance

5.2 Limitations

- **Cached JavaScript Files from OpenAM 12.0.0 May Cause Redirect to undefined:8080**. If you configure an OpenAM 12.0.0 instance with long-lived cache times for the `/XUI/index.html` file, you may experience unexpected redirects to `undefined:8080` after upgrading to OpenAM 13.

To work around this issue, in your chosen web container, or proxy server, reconfigure the cache time for the `/XUI/index.html` file to be short-lived, for example, 5 minutes. Allow enough time that cached files with the long-lived cache time will have expired before upgrading to OpenAM 13.

Note

This issue does not affect upgrades from OpenAM 12.0.1 or later. OpenAM 12.0.1 and later set a short-lived `cache-control` header on UI files to work around the problem of having stale files cached locally.

- **RADIUS Service Only Supports Commons Audit Logging**. The new RADIUS service only supports the new Commons Audit Logging, available in this release. The RADIUS service cannot use the older Logging Service, available in releases prior to OpenAM 13.0.0.
- **Administration Console Access Requires the `RealmAdmin` privilege** . In OpenAM 13, administrators can use the OpenAM administration console as follows:
  - Delegated administrators with the `RealmAdmin` privilege can access full administration console functionality within the realms they can administer. In addition, delegated administrators in the Top Level Realm who have this privilege can access OpenAM's global configuration.
  - Administrators with lesser privileges, such as the `PolicyAdmin` privilege, can not access the OpenAM administration console.
  - The top-level administrator, such as `amadmin`, has access to full console functionality in all realms and can access OpenAM's global configuration.
- **Do Not End Policy Names with a "/" Character**. Do not use a "/" character at the end of a policy name as it will cause OpenAM to not read, edit, or delete the policy.

  After upgrade, users who have policies with a trailing slash "/" character at the end of a policy name should remove the slash (OPENAM-5400). ways:

  To remove slashes in the policy names, remove them as recommended in: OPENAM-5187.
- **Upgrade Incorrectly Sets Default Value for the REST APIs Service**. The workaround is to manually set the default version setting in the REST APIs service to the preferred value:

```
$ openam/bin/ssoadm set-attr-defs -s RestApisService -t Global \
     -a openam-rest-apis-default-version=Latest -u amadmin -f .pass
```

For background information, see OPENAM-6302.

- **OAuth2 Scopes Behavior Affected by Upgrade**. After an upgrade, OAuth 2.0 scope behavior uses a deprecated implementation class, `org.forgerock.openam.oauth2.provider.impl.ScopeImpl`.

  The workaround is to manually update the OAuth 2.0 providers to use the `org.forgerock.openam.oauth2.OpenAMScopeValidator`.

  For background information, see OPENAM-6319.

- **Different OpenAM Version within a Site**. Do not run different versions of OpenAM together in the same OpenAM site.

- **Avoid Use of Special Characters in Policy or Application creation**. Do not use special characters within policy, application or referral names (for example, "my+referral") using the Policy Editor or REST endpoints as OpenAM returns a 400 Bad Request error. The special characters are: double quotes ("), plus sign (+), command (,), less than (<), equals (=), greater than (>), backslash (\), and null (\u0000). (OPENAM-5262)

- **Avoid Using REST Endpoint Names for Realm Names**. Do not use the names of OpenAM REST endpoints as the name of a realm. The OpenAM REST endpoint names that should not be used includes: "users", "groups", "realms", "policies" and "applications". (OPENAM-5314)

- **Deploying OpenAM on Windows in an IPv6 Network**. When deploying OpenAM components on Microsoft Windows in an IPv6 environment, you must use the Java 7 Development Kit on Windows (due to JDK-6230761, which is fixed only in Java 7).

- **Database Repository Type is Experimental**. The Database Repository type of data store is experimental and not supported for production use.

- **Enforcing Session Quotas with Session Failover**. By default OpenAM does not enforce session quotas when running in Site mode without session failover. To work around this behavior, set the server configuration property `openam.session.useLocalSessionsInMultiServerMode=true`. You can set this property in OpenAM console under Configuration > Servers and Sites > Servers > Server Name > Advanced.

- **XACML Policy Import and Export**. OpenAM can only import XACML 3.0 files that were either created by an OpenAM instance, or that have had minor

manual modifications, due to the reuse of some XACML 3.0 parameters for non-standard information.

- **Custom Profile Attributes Are Visible in the User Profile Only With the Classic UI**. The ability to view and edit custom profile attributes is limited to the classic UI. Custom profile attributes do not appear in the user profile when users log in to OpenAM using the XUI.

5.3 Known Issues

The following important known issues remained open at the time release 13 became available. For details and information on other issues, see the OpenAM issue tracker.

5.3.1 Known Issues in OpenAM 13.0.0

The following important issues remained open when OpenAM 13.0.0 became available:

- OPENAM-8142: OAuth2 Access Tokens are inaccessible if the OAuth2 Client contains a space in their name
- OPENAM-8125: IE: can't create policy resource
- OPENAM-8111: User Self Service is active if configured however not turned on in the realm
- OPENAM-8077: XUI does not overwrite stateless session on session upgrade
- OPENAM-8058: ForgeRock Authenticator settings cannot be changed when in production
- OPENAM-7939: Audit file retention "Minimum Free Space Required" field doesn't stop growing the occupied disk space
- OPENAM-7781: persistent cookie auth module does not allow to change cookie name by default
- OPENAM-7746: Authentication in sub-realm fails if DNS alias is used and persistence can not be guaranteed
- OPENAM-7282: Forgotten password submit button is disabled when using autocomplete
- OPENAM-7255: XUI "Delete Label" is inactive for orphan labels
- OPENAM-7069: Shared resources with a user are visible by another valid user

- OPENAM-7054: RADIUS Server Does not Start After Initial Install Without a Web Container Restart
- OPENAM-7035: OAuth2ProviderSettings are not updated if configuration of baseUrlSource service is changed
- OPENAM-7023: UMA resource XUI page does not show permissions from policy
- OPENAM-7002: The email attribute property defined in the email service is not used when sending e-mail in forgotten password flow
- OPENAM-6739: Creating UMA policy as amadmin doesn't show in user's resources
- OPENAM-6666: Re-shared resource that is revoked by resource owner, re-shared user still has access
- OPENAM-6426: Forgot password doesn't print an audit log
- OPENAM-6262: Admin console generates incorrect goto URLs when behind reverse proxy
- OPENAM-2911: IdP initiated SSO with persistent identifier causes URLNotFoundException: Invalid service host name.
- OPENAM-2200: json/sessions/?_queryID=all only returning 120 sessions

Chapter 6

How to Report Problems or Provide Feedback

If you have questions regarding OpenAM which are not answered by the documentation, there is a mailing list which can be found at https://lists.forgerock.org/mailman/listinfo/openam where you are likely to find an answer.

If you have found issues or reproducible bugs within OpenAM 13, report them in https://bugster.forgerock.org.

When requesting help with a problem, include the following information:

- Description of the problem, including when the problem occurs and its impact on your operation
- Description of the environment, including the following information:
  - Machine type
  - Operating system and version
  - Web server or container and version
  - Java version
  - OpenAM version
  - Any patches or other software that might be affecting the problem

- Steps to reproduce the problem
- Any relevant access and error logs, stack traces, or core dumps

Chapter 7

Documentation Updates

The following table tracks changes to the documentation set following the release of OpenAM 13:

Table 7.1. Documentation Change Log

Date	Description
2016-07-15	• Added information about implementing the key rollover feature to Chapter 12, in the *OpenAM Administration Guide*. Key rollover lets you specify multiple encryption and signing keys for SAML providers. • Added a new row, Core Token Service Demand, to Table 9.1, “Impact of Deploying OpenAM Using Stateful and Stateless Sessions” in the *OpenAM Administration Guide*. • Corrected the description of the Auto Federation Attribute property in Section 12.4.8.2, “Hints for Assertion Processing” in the *OpenAM Administration Guide*. • Added a warning not to allow `Content-Type` headers to CORS filters to Section 1.7, “Enabling CORS Support” in the *OpenAM Installation Guide*.

Date	Description
	• Fixed Section 7.4, “Getting Source Code for Sample Mobile Applications” in the *OpenAM Administration Guide* so that it references OpenAM's git repository rather than subversion. OpenAM changed from subversion to git prior to the release of OpenAM 13. • Added the new `org.forgerock.openam.redirecturlvalidator.maxUrlLength` property to Section 1.5, “Servers and Sites Configuration” in the *OpenAM Reference*. • Clarified in Section 2.2, “Java Requirements” that OpenAM requires a JDK installation on the host running the OpenAM web container. • The procedure to turn off user data caching has a new step to disable persistent search. See Procedure 25.1, “To Turn Off Global User Data Caching” in the *OpenAM Administration Guide*.
2016-04-20	• Section 12.4.12, “Configuring Salesforce CRM as a Remote Service Provider” in the *OpenAM Administration Guide* has been updated to reflect changes to the Salesforce CRM user interface. • Descriptions of the Relay State URL List property in Section 12.4.7, “Modifying an Identity Provider's Configuration” in the *OpenAM Administration Guide* and Section 12.4.8, “Modifying a Service Provider's Configuration” in the *OpenAM Administration Guide* have been corrected. • The language code for Japanese in Procedure 5.3, “To Copy the Pages to Customize For the Top-Level Realm” in the *OpenAM Installation Guide* has been corrected. • Procedure 5.5, “To Customize Files You Copied” in the *OpenAM Installation Guide* has been corrected to reflect the changes that occur to the French login page after customization. • Procedure 1.6, “To Prepare OpenAM for JBoss and WildFly” in the *OpenAM Installation Guide* now includes explicit information about the location of the `bootstrap.properties` file.

Date	Description
	• Section 2.4.5.5, “Core - Security” in the *OpenAM Administration Guide* now includes descriptions of the Persistent Cookie Encryption Certificate Alias and Organization Authentication Signing Secret properties. • Chapter 17, in the *OpenAM Administration Guide* and Chapter 7, in the *OpenAM Developer's Guide* have been updated to include the SOAP Security Token Service (STS), and REST STS features introduced in OpenAM 13. These chapters are major rewrites that now include a conceptual overview of SOAP and REST STS, descriptions of all STS configuration properties, instructions for deploying SOAP STS instances, and programmatic and command-line examples of many STS operations. Section 5.10, “Configuring SOAP STS Agents” in the *OpenAM Administration Guide* has been added to describe SOAP STS agent configuration properties.
2016-04-13	• Section 6.7, “Console Ajax JSP Endpoints” in the *OpenAM Reference* has been updated with a corrected text for `FileUpload.jsp`. • Section 3.1, “About the Example Topology” in the *OpenAM Deployment Planning Guide* has been updated with a corrected diagram.
2016-04-05	• Section 2.4.1, “Hints for the Active Directory Authentication Module” in the *OpenAM Administration Guide* and Section 2.4.14, “Hints for the LDAP Authentication Module” in the *OpenAM Administration Guide* has been updated with the property `openam-auth-ldap-operation-timeout`.
2016-04-04	• Section 6.2.4, “CTS Index Import and Build” in the *OpenAM Installation Guide* has been updated with a change in how CTS indexes are created.
2016-02-12	• Added Section 1.3, "Security Advisories" to the OpenAM Release Notes. • Updated the OpenAM Public API Javadoc.

Date	Description
	The previously published Javadoc was generated incorrectly, and did not reflect the supported public API.
2016-01-27	• Initial release of OpenAM 13.0.0 documentation.

Chapter 8

Support

You can purchase OpenAM support subscriptions and training courses from ForgeRock and from consulting partners around the world and in your area. To contact ForgeRock, send mail to info@forgerock.com. To find a partner in your area, see http://forgerock.com/partners/find-a-partner/.