OpenAM Release Notes

Version 12.0.0

Mark Craig
Mike Jang
Vanessa Richie

ForgeRock AS
33 New Montgomery St.,
Suite 1500
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 12.0.0

OpenAM 12.0.0 fixes a number of issues, and provides the following additional features.

1.1 Major New Features

New Features for Users

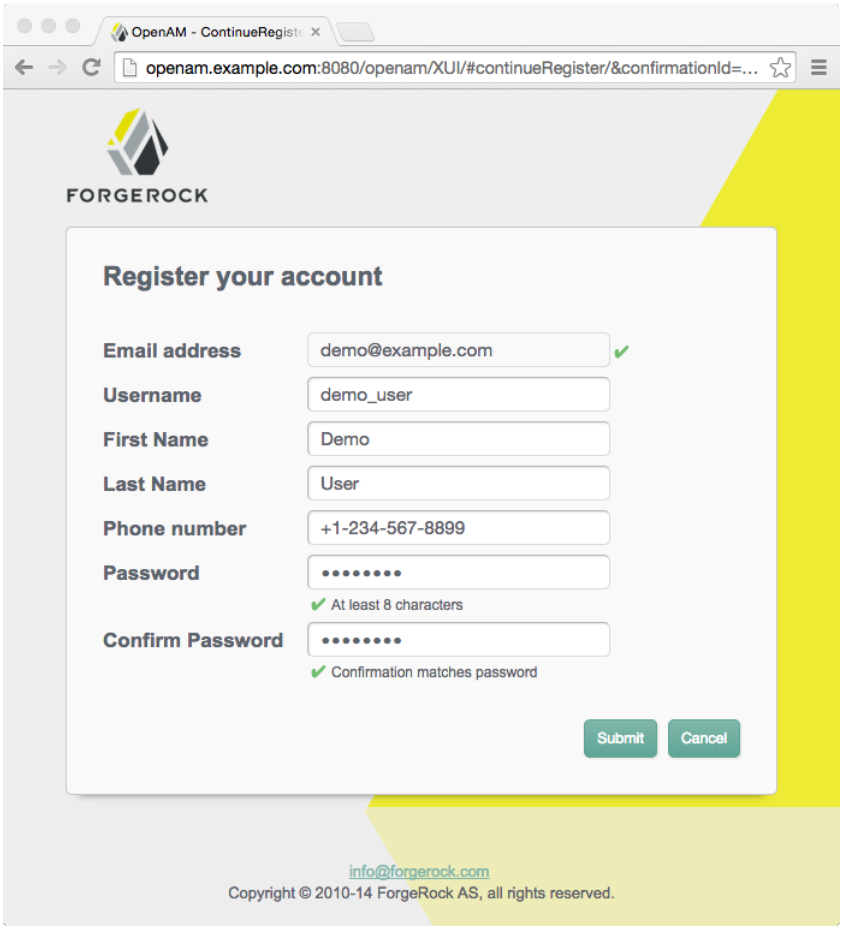
- **User Self-Service**

  OpenAM supports self-service user registration, device management and password reset - reducing costs and increasing customer satisfaction.

  Self-Service User Registration

  OpenAM now offers a user self-registration service through the XUI interface. Click the Register link on the Login page and enter an email address.

  OpenAM will email you to confirm your address, and include a link to a page where you can register your details, as shown below.



For more information, see the *Administration Guide* section *Configuring User Self-Registration*.

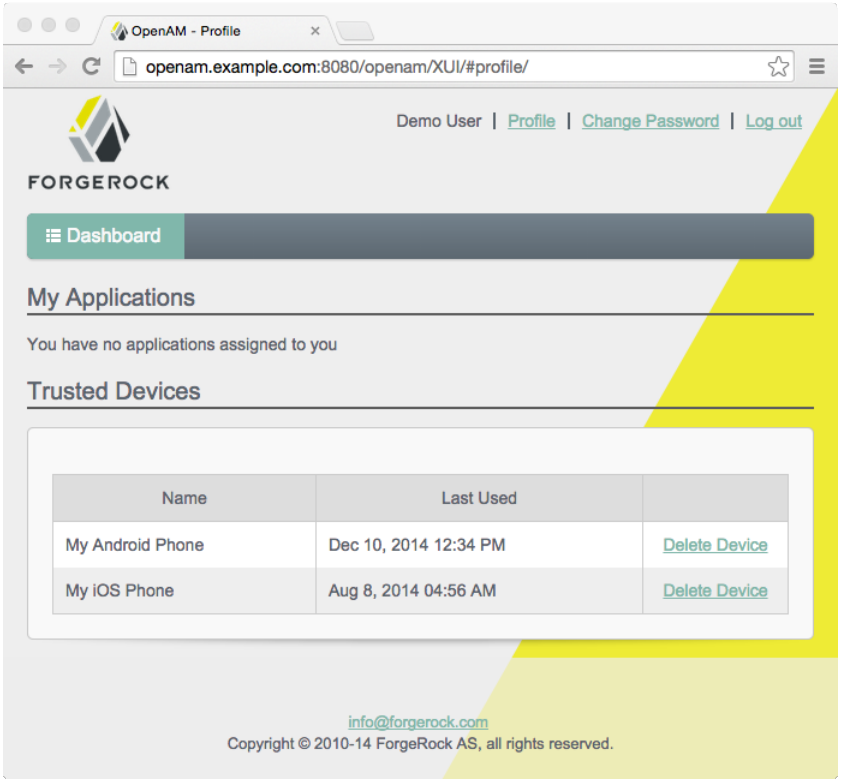
Trusted Device Management

OpenAM allows you to manage a list of trusted devices from your Dashboard page.

When you log in to the console, OpenAM determines if the device you are using differs from that in a stored profile. If there are differences, you will be asked to enter a one-time password.

After the one-time password is verified, you can provide a name for the device and add it to the list of trusted devices.

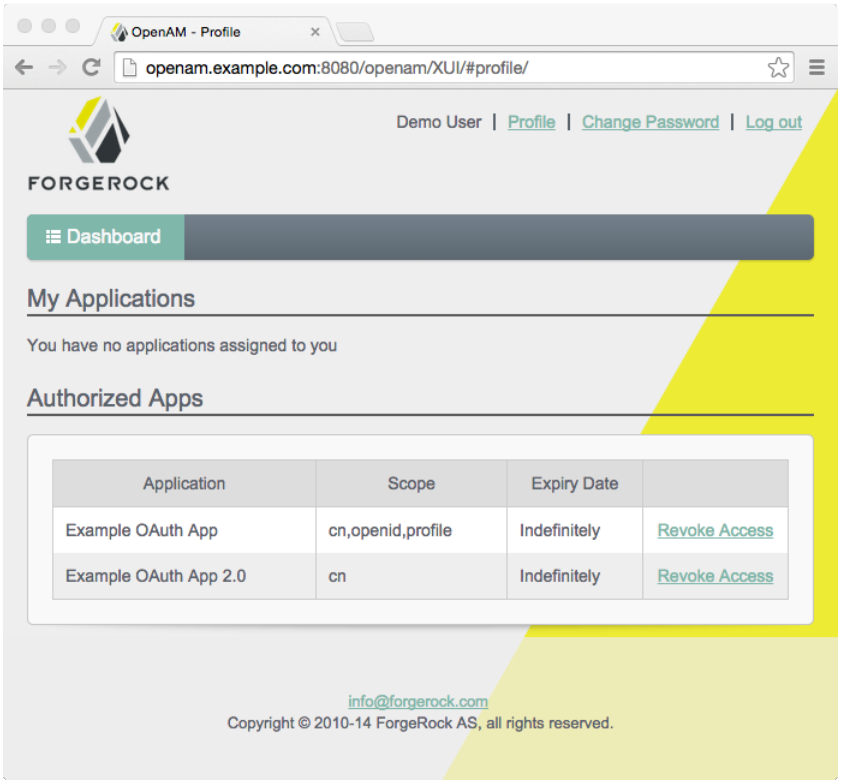
Trusted devices appear in the Dashboard when you log in, as shown below, and can be removed by clicking Delete Device.



For more information, see the *Administration Guide* sections *Hints for the Device ID (Match) Authentication Module* and *Hints for the Device ID (Save) Authentication Module*.

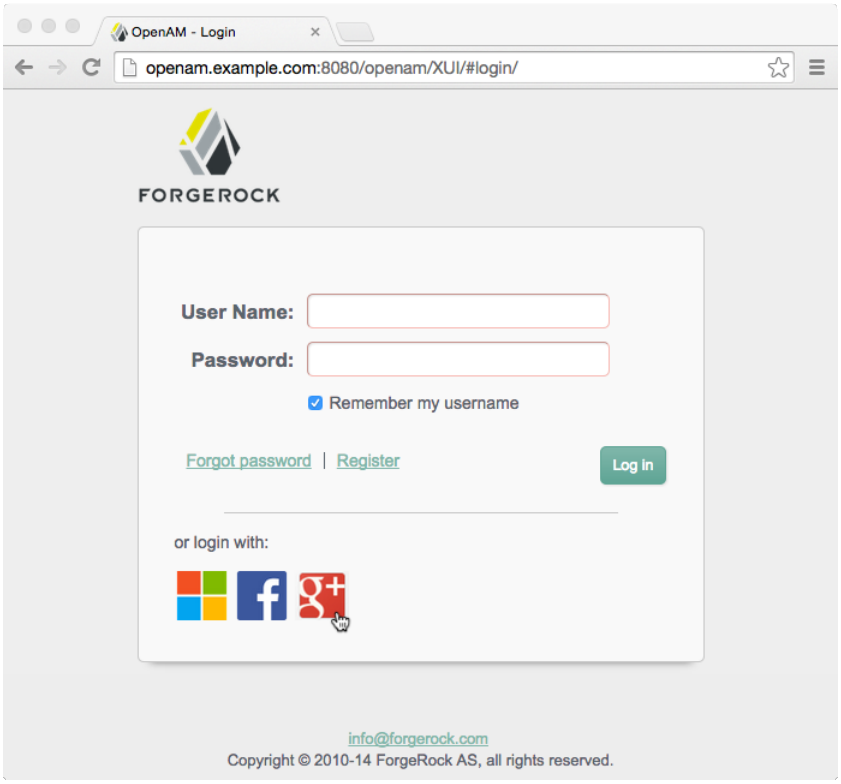
Authorized Application Management

You can now manage your authorized applications (those that use OAuth 2.0 tokens) from the Dashboard link on the user page of the OpenAM console. In the Authorized Apps section, view your OAuth 2.0 tokens or remove them by clicking the Revoke Access link.



- **Social Authentication**

  Log in to an OpenAM protected resource by using your existing social website credentials. OpenAM supports Facebook, Google, Microsoft, or any other OpenID Connect 1.0 compliant identity provider.



The OpenAM administration console provides wizards for quickly configuring social authentication. For more information, see the *Administration Guide* section *Configuring Social Authentication*.

New Features for Administrators

- **New Policy Editor**

  OpenAM policy configuration now supports applications. OpenAM applications act as templates for all the policies that govern access to the protected resources in your applications.

  When you create or edit a policy in OpenAM console for a particular realm, you first choose the application that the policy belongs to, and then create the policy or choose the policy to edit.

The new policy editor user interface allows you to quickly create applications and complex authorization policies, using point-and-click and drag-and-drop operations.

For more information, see the *Administration Guide* chapter *Defining Authorization Policies*

- **Policy Export and Import**

  You can import and export policies to and from XACML 3.0 format files. Use the files for version control, or migration between OpenAM test and production instances, for example.

  For more information, see the *Administration Guide* section *Importing and Exporting Policies*

- **Extended Authorization Subjects**

  You can now choose between an SSO token and an OpenID Connect ID token as the subject to evaluate authorization policies against. OpenID Connect ID Tokens can be used when there is no current user session, for example when an offline batch processing routine acts on behalf of a user.

  For more information, see the *Administration Guide* section *Hints for the OpenID Connect id_token bearer Module*

- **Scripted Authentication Modules**

  You can create custom authentication scripts to gather knowledge about a user to help determine their authentication path. A scripted authentication module

runs a script to perform authentication, making it easier than ever before to develop custom authentication modules.

For example your script could make a call to a third-party proofing service to determine risk, and require stronger authentication depending on the context of the login request.

Scripted authentication modules have access to the same data as other modules in the chain, can access user profile data during authentication, can make HTTP calls to external services, and are sandboxed for more secure operation. The scripts are stored in OpenAM configuration data, and so transparently available across OpenAM Sites. Server-side scripts can be written in either Groovy or JavaScript.

For details on writing authentication module scripts, see the *Developer Guide* chapter *Scripting Authentication*.

For details on configuring scripted authentication modules, see the *Administration Guide* section *Hints For Scripted Authentication Modules*.

- **Scripted Device Identification Modules**

  OpenAM 12.0 introduces new Device ID (Match) and Device ID (Save) authentication modules that support the ability to customize your device fingerprinting implementations.

  The Device ID (Match) Authentication Module uses the new JavaScript/Groovy scripting engine, and demonstrates how scripted modules can be used to add contextual intelligence to the login process.

  For more information, see the *Administration Guide* section *Hints for the Device ID (Match) Authentication Module*

New Features for Developers

- **REST STS for Token Transformation**

  Use the RESTful Security Token Service (REST STS) to convert tokens in the various formats that OpenAM supports, such as OpenID Connect, X.509, and SSO, into a SAML2 token. Given the variety of token types in use today, it can be helpful to have a configurable service that transform tokens.

  For more information, see the *Administration Guide* chapter *The RESTful Security Token Service*

- **OAuth 2.0 and OpenID Connect 1.0 Improvements**

Make use of improved support of the OAuth 2.0 and OpenID Connect 1.0 standards, widely used in mobile and web applications. OpenAM rigorously enforces these standards, improving interoperability, and shortening development lead times.

For more information, see the *Developer's Guide* chapter *RESTful OAuth 2.0 and OpenID Connect 1.0 Services*

OpenAM also supports the *JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants* (OPENAM-4394). This profile allows OAuth 2.0 clients to use JWTs for authentication and to request access tokens. For more information, see the *Administration Guide* section *Authorization*.

- **Fine-Grained Policy APIs**

  Author sophisticated authorization policies by using OpenAM's RESTful interfaces. Manage realms, applications, and policies, list application, condition, and subject types, and request policy decisions using the API, simplifying your applications and deployment.

  For more information, see the *Developer's Guide* chapter *RESTful Authorization and Policy Management Services*

- **GSMA Mobile Connect Support**

  OpenAM now includes support for GSMA Mobile Connect, a profile of OpenID Connect 1.0.

  Mobile Connect lets you authenticate with a mobile phone, regardless of the service or the device on which it is consumed. This allows mobile network operators to serve as identity providers for their customers.

  For more information, see the *Administration Guide* section *Using OpenAM with Mobile Connect*.

- **REST API Versioning**

  OpenAM now assigns REST API features version numbers, to help with backwards-compatibility. You can specify the required version to use when making a call.

  Use the versioning to insulate your REST clients against breaking changes when upgrading an OpenAM instance.

  For more information, see the *Developer's Guide* section *REST API Versioning*.

- **Support for the Latest Platforms**

  OpenAM supports the latest platforms such as Java 8 and Apache Tomcat 8.

For more information on OpenAM requirements and supported versions, see Chapter 2, *Before You Install OpenAM 12.0.0 Software*.

1.2 Additional New Features

- **Audit Logging to Syslog**

  OpenAM now supports logging audit messages to syslog.

  For more information, see the *Administration Guide* section *Audit Logging to syslog* in the *Administration Guide*.

- **Persistent Cookie from Client IP Issued**

  The Persistent Cookie module has been enhanced to be able to enforce that the persistent cookie can only be used from the same client IP to which the cookie was issued.

- **CORS Support for OpenAM APIs**

  OpenAM now supports cross-origin resource sharing (CORS) to allow requests to be made across domains from user agents. Applications in browsers that support CORS can therefore now successfully make calls to an OpenAM server that runs in a different domain from the application.

  Instead, you must configure CORS support in OpenAM's deployment descriptor. For instructions, see the *Installation Guide* section *Enabling CORS Support*.

- **Session Failover Across Sites**

  OpenAM now allows session failover across OpenAM Sites. In order to take advantage of this capability, you must make sure that the underlying Core Token Service (CTS) replicates session data across your OpenAM Sites.

  For details on setting up the underlying Core Token Service, see the *Installation Guide* chapter *Configuring the Core Token Service*.

- **Reduced Cross-Talk**

  OpenAM now attempts to locate a user's session in the Core Token Service (CTS) store before making a crosstalk request through a back channel to other OpenAM servers in the cluster.

  The reduction in network traffic can increase performance.

For more information, see the *Install Guide* section *To Configure Site Load Balancing*.

- **Asynchronous Core Token Service Requests**

  A change to the Core Token Service (CTS) means that requests are no longer performed synchronously. CTS processes all requests asynchronously in the background, allowing callers (that is, those entities that call CTS) to send subsequent requests without waiting for a previous request to finish processing, improving response times and performance.

  For more information, see the *Install Guide* section *CTS Tuning Considerations*.

- **Fine-Grained Settings for LDAP Connections**

  OpenAM now provides additional options for tuning LDAP connection pool sizes and timeouts related to the Core Token Service and to other components that use LDAP connections. For more information, see the *Administration Guide* section *Tuning LDAP CTS & Configuration Store Settings*.

- **REST Policy Filter Rules**

  OpenAM now supports REST Policy Filter rules that simplify the configuration to protect ForgeRock common REST APIs.

- **OAuth 2.0 Scope Conditions**

  OpenAM now supports an OAuth2 Scope condition that lets the you set required OAuth 2.0 scopes as a policy condition.

- **Configurable DN Cache for LDAP Data Stores**

  OpenAM now has the capability to enable and disable DN caching. DN caching helps avoid DN lookups that can happen in bursts during authentication. (OPENAM-3822).

- **Quicker UI Customization**

  While customizing the UI, you can set the advanced server property, `org.forgerock.openam.core.resource.lookup.cache.enabled`, to `false` to allow OpenAM immediately to pick up changes to the files as you customize them (OPENAM-3989). You can set advanced server properties in OpenAM Console under Configuration > Servers and Sites > *`Server Name`* > Advanced. For production servers, leave this set to the default, `true`.

- **Whitelist for Custom Login URIs**

OpenAM now includes a property that specifies a whitelist for custom login URIs so that the CDCServlet and the Distributed Authentication UI (DAS) can check login URI values against those in the whitelist.

The property name is `org.forgerock.openam.cdc.validLoginURIs`. For more information about this property, see the *Reference* section on advanced properties, *Servers > Advanced*.

- **OpenID Connect Registration Without an Access Token**

  OpenAM can now be configured to let OpenID Connect clients register dynamically without having to provide an access token (OPENAM-3604). For details, see the documentation on the advanced server property, `org.forgerock.openam.openidconnect.allow.open.dynamic.registration`, in the *OpenAM Reference* section, *Servers > Advanced*.

- **Policy Support for Common HTTP Operations**

  OpenAM policies now let you allow and deny not only HTTP GET and HTTP POST, but also HTTP DELETE, HEAD, OPTIONS, PATCH, and PUT (OPENAM-336).

- **REST Logging**

  OpenAM now supports audit logging and debug notifications for any request going to a common REST (CREST) endpoint. OpenAM audits every request going to any CREST endpoint and writes to two files: `amRest.access` and `amRest.authx`.

  The `amRest.access` file records all accesses to a CREST endpoint (except `/authenticate`), regardless of whether the request successfully reached the endpoint through policy authorization.

  The `amRest.authz` file records all CREST authorization results regardless of success. If a request has an entry in the `amRest.access` log, but no corresponding entry in `amRest.authz`, then that endpoint was not protected by an authorization filter and therefore the request was granted access to the resource.

  OpenAM now provides additional information in its debug notifications depending on the message type (error, warning or message) including realm, user, and result of the operation.

  For more information on CREST logging, see *Logging*.

Chapter 2

Before You Install OpenAM 12.0.0 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.

- CentOS 6, 7
- Microsoft Windows Server 2008, 2008 R2, 2012, 2012 R2
- Oracle Linux 6, 7
- Oracle Solaris x64 10, 11
- Oracle Solaris SPARC 10, 11
- Red Hat Enterprise Linux 6, 7
- SuSE Linux 11

- Ubuntu Linux 12.04 LTS, 14.04 LTS

2.2 Java Requirements

OpenAM server software runs in a Java EE Web container, and requires a Java Development Kit.

ForgeRock supports customers using the following Java versions. ForgeRock recommends the most recent Java update, with the latest security fixes.

- Oracle Java Development Kit 6, 7, or 8
- IBM Java Development Kit 6 or 7 (when deploying in WebSphere only)

2.3 OpenAM Web Application Container Requirements

ForgeRock supports customers using OpenAM server software in the following web application container versions.

- Apache Tomcat 6, 7, 8 (ForgeRock's preferred web container for OpenAM)
- IBM WebSphere Application Server 8, 8.5
- JBoss Enterprise Application Platform 6
- JBoss Application Server 7
- Oracle WebLogic Server 11g, 12c

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

2.4 Data Store Requirements

The following table summarizes OpenAM data store support.

Table 2.1. Supported Data Stores

Data Store	Versions	Core Token Service (CTS) Data Store	Configuration Data Store	User Data Store
Embedded OpenDJ (included in OpenAM)	2.6.2	Supported	Supported	Supported

Data Store	Versions	Core Token Service (CTS) Data Store	Configuration Data Store	User Data Store
External OpenDJ	2.6, 2.6.2	Supported	Supported	Supported
IBM Tivoli Directory Server	6.3			Supported
Microsoft Active Directory	2008, 2008 R2, 2012, 2012 R2			Supported
Oracle Directory Server Enterprise Edition	11g	**NOT SUPPORTED**	Supported When using DSEE as a configuration store, you must set up an external OpenDJ directory service as a Core Token Service data store as well, and you must configure OpenAM to use the external OpenDJ directory service as the CTS data store.	Supported
Oracle Unified Directory	11g		Supported	Supported

2.5 Browser Requirements

The following table summarizes browser support.

Table 2.2. Supported Platforms & Browsers

Client Platform	Chrome 16 or later	Internet Explorer 9 or later	Firefox 3.6 or later	Safari 5 or later
Apple iOS 7 or later	Supported			Supported
Apple Mac OS X 10.8 or later	Supported		Supported	Supported

Client Platform	Chrome 16 or later	Internet Explorer 9 or later	Firefox 3.6 or later	Safari 5 or later
Google Android 4.3 or later	Supported			
Microsoft Windows 7 or later	Supported	Supported	Supported	Supported
Ubuntu Linux 12.04 LTS or later	Supported		Supported	

2.6 Native Application Platform Requirements

ForgeRock supports customers' use of OpenAM REST and other client APIs in native applications on the following platforms.

- Apple iOS 7 or later
- Apple Mac OS X 10.8 or later
- Google Android 4.3 or later
- Microsoft Windows 7 or later
- Ubuntu Linux 12.04 LTS or later

Other combinations might work as well. When opening a support ticket for an issue, however, make sure you can also reproduce the problem on one of these platforms.

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

OpenAM Changes & Deprecated Functionality

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

3.1 Important Changes to Existing Functionality

- All OpenAM core server, tools, and agent installers now display a software license acceptance screen prior to configuration. You must agree to the license to continue the configuration.

  For users implementing scripted or silent installs, the installers and upgrader tools provide a `--acceptLicense` command-line option, indicating that you have read and accepted the terms of the license. If the option is not present on the command-line invocation, the installer or upgrader will interactively present a license agreement screen to the user.

- When you visit the Policies tab for a realm in OpenAM console, OpenAM now directs you to the new policy editor. For instructions on using the new policy editor, see the *Administration Guide* chapter, *Defining Authorization Policies*. Notice that policies now belong to applications as described in that chapter.

  OpenAM has changed its internal representation for policies to better fit the underlying implementation, which is based on a new engine designed for higher performance and finer grained policies. When you upgrade to this version, OpenAM migrates your policies to the new representation.

Depending on your existing policies before upgrade, you can see the following differences:

- Existing policies with multiple resource rules map to multiple new policies.

  When a single policy maps to multiple policies during migration, OpenAM appends a number to the existing name for each new policy. This allows you to recognize the set of policies when you must manage them together, for example to change them all in the same way.

  This behavior is to optimize policy evaluation performance. The newer policy engine matches resources to policies during evaluation with indexing that proves very efficient as long as each policy specifies one resource pattern. OpenAM processes the list of resources in policies in linear fashion, so long lists of resources can slow policy evaluation.

- Conditions in existing policies map to newer representations.

  New representations exist for all existing conditions provided in OpenAM out of the box. Custom conditions developed for your deployment do not map to any of the newer conditions provided. In that case you must implement your custom conditions by implementing the newer service provider interfaces, and then replace your existing policies to use them.

  To see how to implement a custom policy plugin, see the *Developer's Guide* chapter, *Customizing Policy Evaluation*.

- When OpenAM encounters issues migrating policies during upgrade, 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.

OpenAM configuration has changed in several ways to accommodate the changes to the way policies are managed:

- The Policy Configuration Service is simplified. For details see the *Reference* section, *Policy Configuration*.

- OpenAM now requires policy referrals only when an application is administered across multiple realms, as can be the case when one policy agent protects multiple applications. Otherwise, OpenAM can use new settings in policy agent profiles to direct policy agent requests to the appropriate realm and application.

Note

Referrals are not shown by default in the policy editor. To enable them, in the OpenAM console, select Configuration > Global > Policy Configuration, set Activate Referrals to Enabled, and then click Save.

The web and Java EE policy agent profiles includes the new settings under OpenAM Services > Policy Client Service in OpenAM console. These new settings allow you to set the realm and application for a policy agent. The settings are compatible with existing policy agents, as they are not used by the policy agents themselves, but instead by OpenAM when handling policy agent requests.

The fix for OPENAM-3509 ensures that OpenAM considers a trailing slash as part of the resource name to match. This improves compatibility between self and subtree modes, and compatibility with older policy agents.

- The Device ID (Match), HMAC One-Time Password (HOTP), and Device ID (Save) modules, configured together in an authentication chain, provide the same functionality as the Device Print Authentication module that is present in OpenAM 11.x versions.

  The Device Print authentication module is only available for OpenAM 11.x versions and their upgrades. If you have upgraded from OpenAM 11.x to OpenAM 12.0 you can still use the Device Print module, customize it, and create new instances of the module or use the Device ID (Match) and Device ID (Save) modules.

  ### Important

  The Device ID (Match) profiles (that is, device fingerprints) are incompatible with profiles created from the Device Print authentication module. If the user has existing device print profiles, created from the Device Print authentication module, these old profiles will always fail to match the client's new device profiles using the scripted Device ID (Match) module even when using the same device.

  With the Device ID (Match) and Device ID (Save) modules, the user must re-save each device profile, which deletes the old 11.x profiles stored for the user.

- Following a change to the SAML 2.0 pages in OpenAM, you no longer customize `saml2login.template` and `saml2loginwithrelay.template` to add a progress bar for single sign on. Instead, customize `saml2/jsp/`

`autosubmitaccessrights.jsp` as described in the procedure, *To Indicate Progress During SSO*.

- Changing passwords by using a PUT REST API call is no longer supported.

  Use a POST request to `/json/`*`subrealm`*`/users/`*`username`*`?_action=changePassword` to change a password.

- The response returned when submitting incorrect credentials to `/json/authenticate` has changed.

Table 3.1. Failed Authentication Message

OpenAM 11.0.1	OpenAM 12.0.0
`{ "errorMessage": "Authentication Failed!!", "failureUrl": "https://openam.example.com:8443" }`	`{ "code": 401, "reason": "Unauthorized", "message": "Authentication Failed!!", "detail": { "failureUrl": "https://openam.example.com:8443" } }`

- When running OpenAM on WebLogic 11g, you must add a WebLogic-specific descriptor file, `WEB-INF/weblogic.xml` to the .war before deployment.
- In the OpenID Connect 1.0 module you can map local user profile attributes to OpenID Connect Token claims, allowing the module to retrieve the user profile based on the ID Token. The key is the ID Token field name and value is the local user profile attribute name. The default has been changed as follows: mail=email, uid=sub. (OPENAM-5263)
- The class hierarchy for the `ResourceName` interfaces has changed. Previous implementations should be source-compatible, but will not be binary-compatible, and will need recompiling.
- The OAuth2 provider uses RSA as its default encryption algorithm. The default OAuth2 client agent configuration has been changed to RS256 to match the OAuth2 provider algorithm. The client agent continues to support HMAC algorithms; only the default encryption algorithm has been changed to support out-of-the-box functionality. (OPENAM-5279)
- The distributed authentication service (DAS) and cross-domain single sign-on (CDSSO) do not support the `iPSPCookie/DProPCookie` query string parameter to set a `DProPCookie` in the user-agent as a mechanism for cookie persistence. Neither DAS nor CDSSO retains `iPSPCookie=yes`.
- Updates to OAuth 2.0 and OpenID Connect authentication modules mean that any custom implementations of `org.forgerock.openam.authentication.modules.`

`oauth2.AccountMapper` or `org.forgerock.openam.authentication.modules.oauth2.AttributeMapper` no longer work, and needs to be reimplemented against `org.forgerock.openam.authentication.modules.common.mapping.AttributeMapper` and/or `org.forgerock.openam.authentication.modules.common.mapping.AccountProvider` as appropriate.

- The XUI, now the default for end-user pages, handles DNS/realm alias differently from the classic UI, which was the default in previous OpenAM versions. With the classic UI, the realm alias is specified both in the host name and the URI path. With the XUI, the host name alone specifies the realm. The XUI evaluates a realm specified in the path of the URL as a subrealm of the realm specified by the host name alias.

  For example, with the classic UI, you could authenticate to a realm, `realm1` using the DNS alias `realm1.example.com:8080` and the realm query parameter, `realm=realm1`, as follows:

  ```
  http://realm1.example.com:8080/openam/UI/Login?realm=realm1
  ```

  With XUI, you do not include a realm in the URI if it has already been mapped as now any URI realm is additive and specifies a subrealm of the DNS alias realm. For example, using the following URL indicates that you are attempting to authenticate to `/realm1/realm1` (that is, the sub-realm, `realm1` under the realm, `realm1`).

  ```
  http://realm1.example.com:8080/openam/XUI/#Login/realm1
  ```

  As another example, if you have a sub-realm called `test` under `/realm1` and make a request to:

  ```
  http://realm1.example.com:8080/openam/XUI/#Login/test
  ```

  The request authenticates to `/realm1/test`. Note also that the use of URI realm is preferred over realm as a query parameter.

3.2 Deprecated Functionality

The following functionality is deprecated in OpenAM 12.0.0, and is likely to be removed in a future release.

- Classic (JATO-based) UI is deprecated for end user pages. OpenAM offers the JavaScript-based XUI as a replacement. Classic UI for end user pages is likely to be removed in a future release.
- Older REST services relying on the following endpoints are deprecated.

  /identity/attributes
  /identity/authenticate
  /identity/authorize
  /identity/create
  /identity/delete
  /identity/isTokenValid
  /identity/logout
  /identity/read
  /identity/search
  /identity/update
  /ws/1/entitlement/decision
  /ws/1/entitlement/decisions
  /ws/1/entitlement/entitlement
  /ws/1/entitlement/entitlements

  The following table shows how legacy and newer endpoints correspond.

Table 3.2. REST Endpoints

Deprecated URIs	**Newer Evolving URIs**
/identity/attributes	/json/users
/identity/authenticate	/json/authenticate
/identity/authorize	/json/policies?_action=evaluate, /json/policies?_evaluateTree
/identity/create, /identity/delete, /identity/read, /identity/search, /identity/update	/json/agents, /json/groups, /json/realms, /json/users
/identity/isTokenValid	/json/sessions/*tokenId*?_action=validate
/identity/logout	/json/sessions/?_action=logout
/ws/1/entitlement/decision, /ws/1/entitlement/decisions, /ws/1/entitlement/entitlement, /ws/1/entitlement/entitlements	/json/policies?_action=evaluate, /json/policies?_evaluateTree
N/A	/json/applications
N/A	/json/applicationtypes
N/A	/json/conditiontypes
N/A	/json/dashboard
N/A	/json/decisionscombiners
N/A	/json/policies
N/A	/json/referrals
N/A	/json/serverinfo
N/A	/json/subjectattributes
N/A	/json/subjecttypes

Deprecated URIs	Newer Evolving URIs
N/A	/xacml/policies

Find examples in the *Developer Guide* chapter *Using RESTful Web Services*.

Support for the older REST services is likely to be removed in a future release in favor of the newer REST services. Older REST services will be removed only after replacement REST services are introduced.

- With the implementation of XACML 3.0 support when importing and exporting policies the following `ssoadm` commands have been replaced:

Table 3.3. Policy Import and Export with ssoadm

Deprecated Command	Newer Evolving Command
`create-policies`	`create-xacml`
`delete-policies`	`delete-xacml`
`list-policies`	`list-xacml`
`update-policies`	`create-xacml`

For more information, see the *OpenAM Reference* section ssoadm — configure OpenAM core services.

- With the implementation of OAuth 2.0 in this release, OAuth 1.0 has been deprecated. OAuth 1.0 support was originally provided in OpenAM 9.
- The Netscape LDAP API is to be removed from OpenAM, with OpenAM using the OpenDJ LDAP SDK instead. This affects all classes in `com.sun.identity.shared.ldap.*` packages.
- OpenAM currently uses Sun Java System Application Framework (JATO). JATO is deprecated and is likely to be replaced in a future release.
- With the implementation of the Persistent Cookie authentication module, the Core Authentication module persistent cookie options are deprecated and are likely to be removed in a future release.
- The OAuth 2.0 plugin interface for custom scopes, `Scope` is deprecated and likely to be removed in a future release.

  Custom OAuth 2.0 scopes plugins now implement the `ScopeValidator` interface instead. For an example, see the *Developer's Guide* chapter, *Customizing OAuth 2.0 Scope Handling*.
- The OAuth 2.0 plugin interface for custom response types, `ResponseType` is deprecated and likely to be removed in a future release.

Custom OAuth 2.0 response type plugins now implement the `ResponseTypeHandler` interface instead.

3.3 Removed Functionality

- No functionality has been removed in this release.

Chapter 4

OpenAM Fixes, Limitations, & 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 12.0.0.

4.1 Key Fixes

The following bugs were fixed in release 12.0.0. For details, see the OpenAM issue tracker.

- OPENAM-4340: Configurator is unable to handle special characters in passwords
- OPENAM-4264: IDPAccountMapper.getNameID() does not receive the SP Entity ID if there is no SPNameQualifier in the SAML request
- OPENAM-4262: IDP Proxy should set destination depending on the Binding
- OPENAM-4236: CookieUtils.addCookieToResponse only sends Max-Age attribute
- OPENAM-4229: Change Password as User does not work using AD-LDS (ADAM) User Store
- OPENAM-4227: Set Password as Administrator does not work using AD-LDS (ADAM) User Store

- OPENAM-4094: OAuth2 Authentication Module does not work, if com.iplanet.am.cookie.encode is true.
- OPENAM-3969: 403 on using /json/<realm>/policies?_action=evaluate
- OPENAM-3822: Datastore authentication fails after modify DN operation.
- OPENAM-3758: OAuth2 save consent when no scope is present is not working
- OPENAM-3731: Sun JDK 1.6.0_43: some requests cause never-ending loop in SAML2Utils.decodeFromRedirect
- OPENAM-3964: The class hierarchy for `ResourceName` interfaces has changed in this issue. Previous implementations should still be source-compatible but are not binary-compatible. You must recompile your custom code if you implemented the `ResourceName` interfaces.
- OPENAM-3640: StackOverFlowError in WebtopNaming
- OPENAM-3573: IDP Initiated federation with missing SPNameQualifier result in exception
- OPENAM-3513: wrong l10n key in code, ssoadm delete-auth-instance fails on error reporting
- OPENAM-3437: RelayState validation fails during SLO
- OPENAM-3428: DJLDAPv3Repo breaks Active Directory when using sAMAccountName as naming attribute with the DN being the CN
- OPENAM-3385: DJLDAPv3Repo Error Unexpected Results Returned when searching Active Directory users from the root
- OPENAM-3314: Hosted IDPs/SPs in COTs with Spaces
- OPENAM-3271: OpenAM Bootstrap file not found for upgrade from 10.0.1 to 11.0.0 if both .openamcfg and .openssocfg exist
- OPENAM-3225: SAML authentication throws NPE with IDP metadata showing certain characteristics
- OPENAM-2532: deleting ActiveDirectory DataStore from subrealm deleting parent's referrals too.
- OPENAM-2464: HOTP auth module sends 2 HOTP codes, if "Request new code" is clicked.
- OPENAM-2322: NULL pointer exception in windowsdesktopsso.java file when doing kerberos service ticket authentication with Openssoclientsdk.jar client program - backward compatibility broken

- OPENAM-1829: .NET Fedlet - "Signature Transform" and "Canonicalization Method" should be configurable
- OPENAM-1789: .NET Fedlet creates SAML2 IDs with incorrect format
- OPENAM-1773: DAS does not handle goto whitelisting
- OPENAM-1755: The .NET fedlet uses invalid constants "True" "False" for some boolean XML attributes
- OPENAM-1749: AttributeQueryUtil.getAttributeMapForFedlet eats non-Success StatusCode from IDP
- OPENAM-1655: AttributeQueryUtil ignores configured SPAttributeMapper
- OPENAM-1058: Enhance to use attribute names defined in the HOTP service config for the telephone, carrier and email address.
- OPENAM-957: Null pointer exceptions in IDPSSOFederate.getAuthnRequest()
- OPENAM-474: Dynamic User Creation not populating all available attributes onto newly created user
- OPENAM-371: Remove frequently occurring meaningless Error stack trace from debug log
- OPENAM-294: ssoadm: create and update
- OPENAM-110: Attribute name comparison in AttributeQueryUtil.isSameAttribute()
- OPENAM-61: SAML2 appliesTo not being HTML character-encoded

4.2 Limitations

- **Different OpenAM Version within a Site**. Do not run different versions of OpenAM together in the same OpenAM site.
- **Deleting Referral Policy**. OpenAM allows you to delete a referral policy even if policies depending on the referral still exist in the target realm. Deleting a referral policy that other policies depend on can cause problems during policy evaluation. You must therefore make sure that no policies depend on any referrals that you delete.
- **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.
- **OpenAM with Embedded Directory Server in IPv6 Networks**. On hosts with pure IPv6 networks, OpenAM configuration has been seen to fail while starting the embedded OpenDJ directory server (OPENAM-3008).
- **JBoss 6.3 Support for Java 8**. As of this writing, JBoss 6.3/AS 7.4.0 does not support Java 8. Until JBoss 6.3 fully supports Java 8, you can use JDK 1.7.0_56 (OPENAM-4876).
- **Note about HttpServletResponse & HttpServletRequest**. The `HttpServletRequest` instance passed to `AMPostAuthProcessInterface#onLogout` will be null. The `HttpServletResponse` instance passed to `AMPostAuthProcessInterface#onLogout` is not the actual `HttpServletResponse` corresponding to the request but a faux instance whose only purpose is to transfer headers back to the actual `HttpServletResponse` (OPENAM-4045).
- **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.

4.3 Known Issues

The following important known issues remained open at the time release 12.0.0 became available. For details and information on other issues, see the OpenAM issue tracker.

- OPENAM-5321: Cross realm session upgrade not handled properly by XUI
- OPENAM-5243: REST HTTP codes are different for some actions in AM 11.0.2 and AM 12
- OPENAM-5237: OAuth2 authorization consent page uses absolute URL in FORM tag
- OPENAM-5234: AuthLevel policy condition does not work with pol. agents when result code 403 is expected
- OPENAM-5183: CTS port settings are reverted to default when doing upgrade from AM 11 to AM 12
- OPENAM-4517: GUI installer crashes and restarts in Safari
- OPENAM-4430: Upgrade wizard is out of date for other languages than EN
- OPENAM-3924: XUI is ignoring iplanet-am-admin-console-password-reset-enabled and requesting user password be entered anytime password is changed
- OPENAM-3466: LDAP authentication module does not apply the change of the password for the bind DN user until restart
- OPENAM-3442: CTS TokenType is missing an index
- OPENAM-3223: Policy Wildcard Matches doesn't work after OpenAM upgrade with an ODSEE
- OPENAM-3109: Token conflicts can occur if OpenDJ servers are replicated
- OPENAM-3056: Retrieving roles may fail when using more than one data store
- OPENAM-3048: RESTful authentication using curl doesn't work on the WebLogic 12c (12.1.1.0)
- OPENAM-2715: Mandatory OAuth2 Provider settings not enforced in the UI
- OPENAM-2705: People container name/value configs are not always correctly used - backport
- OPENAM-2656: PrefixResourceName#compare() strips off trailing '/' in PathInfo
- OPENAM-2608: Restricted Token validation does not work in legacy REST API
- OPENAM-2564: resource-based authentication with DistAuth not working
- OPENAM-2537: SAML AuthContext mapper auth level setting inconsistencies

- OPENAM-2469: IdP initiated SSO endpoints should honor RelayState even when they're POSTed
- OPENAM-2453: HTTP GET /ws/1/entitlement/privilege? HTTP 400 with message "Unable to search privileges."
- OPENAM-2404: new_org.jsp is displayed from the original realm in case of session upgrade
- OPENAM-2168: Authentication Success Rate and Authentication Failure Rate are always 0
- OPENAM-2137: DSConfigMgr can hide exception root causes
- OPENAM-2085: Unreliable policy evaluation results with com.sun.identity.agents.config.fetch.from.root.resource enabled
- OPENAM-2023: Federation Connectivity Test fails with Account termination is not working
- OPENAM-1946: Password change with AD does not work when old password is provided
- OPENAM-1945: Default Configuration create invalid domain cookie
- OPENAM-1892: Only Accept certificate for authentication if KeyUsage is correct
- OPENAM-1886: Session invalidated on OpenAM server is not deleted from SFO datastore
- OPENAM-1852: Oauth2 auth-module can not be used with DistAuth
- OPENAM-1831: OpenAM 10.0 subrealm DNS alias doesn't work as expected unless setting com.sun.identity.server.fqdnMap
- OPENAM-1811: DAS response serialization is not working as expected when using PAP
- OPENAM-1660: Read-access to SubjectEvaluationCache is not synchronized
- OPENAM-1659: Default Authentication Locale is not used as fallback
- OPENAM-1505: LogoutViewBean does not use request information for finding the correct template
- OPENAM-1456: Change of the agent group in the J2EE policy agent profile causes profile corruption
- OPENAM-1323: Unable to create session service when no datastore is available

- OPENAM-1317: With ssoadm create-agent, default values are handled differently for web agents and j2ee agents
- OPENAM-1269: Entitlements are incorrectly converted to policies
- OPENAM-1237: Property 'noSubjectKeyIdentifier' is missing in fmWSSecurity.properties
- OPENAM-1219: SAML 2 metadata parsing breaks in glassfish 3.1.2
- OPENAM-1194: Unable to get AuthnRequest error in multiserver setup
- OPENAM-1181: Improperly defined applications cause the policy framework to throw NPE
- OPENAM-1137: Error message raised when adding a user to a group
- OPENAM-1111: Persistent search in LDAPv3EventService should be turned off if caching is disabled
- OPENAM-1105: Init properties sometimes don't honor final settings
- OPENAM-774: Invalid characters check not performed.
- OPENAM-291: SelfWrite permissions are denied to sub realms
- OPENAM-71: SAML2 error handling in HTTP POST and Redirect bindings

Chapter 5

How to Report Problems & 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 12.0.0, 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 6

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/.