OpenAM Java EE Policy Agent Reference

MarkCraig
VanessaRichie
MikeJang

,
, ,

Abstract

Guide to installing OpenAM Java EE policy agents. OpenAM provides open source Authentication, Authorization, Entitlement and Federation software.

Admonition graphics by Yannick Lung. Free for commerical use. Available at Freecns Cumulus.

Table of Contents

Chapter 1

Java EE Agent Configuration Properties

Java EE Agents use the following configuration properties. Bootstrap properties are always configured locally, whereas other agent configuration properties are either configured centrally in OpenAM or locally using the agent properties file.

1.1 Bootstrap Configuration Properties

These properties are set in `config/`.

`am.encryption.pwd`

When using an encrypted password, set this to the encryption key used to encrypt the agent profile password.

`com.iplanet.am.naming.url`

Set this to the naming service URL(s) used for naming lookups in OpenAM. Separate multiple URLs with single space characters.

`com.iplanet.am.service.secret`

When using a plain text password, set this to the password for the agent profile, and leave `am.encryption.pwd` blank.

When using an encrypted password, set this to the encrypted version of the password for the agent profile. Use the command **./agentadmin --encrypt *`agentInstance`* *`passwordFile`*** to get the encrypted version.

`com.iplanet.am.services.deploymentDescriptor`

Set this to the URI under which OpenAM is deployed, such as `/openam`.

`com.iplanet.services.debug.directory`

Set this to the full path of the agent's debug log directory where the agent writes debug log files.

`com.sun.identity.agents.app.username`

Set this to the agent profile name.

`com.sun.identity.agents.config.local.logfile`

Set this to the full path for agent's audit log file.

`com.sun.identity.agents.config.lock.enable`

Set this to `true` to require an agent restart to allow agent configuration changes, even for hot-swappable parameters. Default is `false`.

`com.sun.identity.agents.config.organization.name`

Set this to the realm name where the agent authenticates to OpenAM.

`com.sun.identity.agents.config.profilename`

Set this to the profile name used to fetch agent configuration data. Unless multiple agents use the same credentials to authenticate, this is the same as `com.sun.identity.agents.app.username`.

`com.sun.identity.agents.config.service.resolver`

Set this to the class name of the service resolver used by the agent.

`com.sun.services.debug.mergeall`

When set to `on`, the default, the agent writes all debug messages to a single file under `com.iplanet.services.debug.directory`.

1.2 Agent Configuration Properties

These properties are set in `config/` if your agent uses local configuration. If your agent uses centralized configuration, the properties are set in OpenAM.

`com.iplanet.am.cookie.name`

Name of the SSO Token cookie used between the OpenAM server and the agent. Default: `iPlanetDirectoryPro`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > Cookie Name.

`com.iplanet.am.sdk.remote.pollingTime`

If notifications are not enabled and set to a value other than zero, specifies the time in minutes after which the agent polls to update cached user management data. Default: 1

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > User Data Cache Polling Time.

`com.iplanet.am.server.host`

Specifies the OpenAM authentication service host name.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > OpenAM Authentication Service Host Name.

`com.iplanet.am.server.port`

Specifies the OpenAM authentication service port number.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > OpenAM Authentication Service Port.

`com.iplanet.am.server.protocol`

Specifies the protocol used by the OpenAM authentication service.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > OpenAM Authentication Service Protocol.

`com.iplanet.am.session.client.polling.enable`

When enabled, the session client polls to update the session cache rather than relying on notifications from OpenAM.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > Enable Client Polling.

`com.iplanet.am.session.client.polling.period`

Specifies the time in seconds after which the session client requests an update from OpenAM for cached session information. Default: 180

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > Client Polling Period.

`com.iplanet.security.encryptor`

Specifies the agent's encryption provider class.

Default: `com.iplanet.services.util.JCEEncryption`

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Miscellaneous > Encryption Provider.

`com.iplanet.services.debug.level`

Default is `Error`. Increase to `Message` or even `All` for fine-grained detail.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > Agent Debug Level.

`com.sun.identity.agents.config.access.denied.uri`

Specifies the URIs of custom pages to return when access is denied. The key is the web application name. The value is the custom URI.

To set a global custom access denied URI for applications without other custom access denied URIs defined, leave the key empty and set the value to the global custom access denied URI, `/sample/accessdenied.html`.

To set a custom access denied URI for a specific application, set the key to the name of the application, and the value to the application access denied URI, such as `/myApp/accessdenied.html`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Resource Access Denied URI.

`com.sun.identity.agents.config.agent.host`

Specifies the host name of the agent protected server to show to client browsers, rather than the actual host name.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Alternative Agent Host Name.

`com.sun.identity.agents.config.agent.port`

Specifies the port number of the agent protected server to show to client browsers, rather than the actual port number.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Alternative Agent Port Name.

`com.sun.identity.agents.config.agent.protocol`

Specifies the protocol used to contact the agent from the browser client browsers, rather than the actual protocol used by the server. Either `http` or `https`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Alternative Agent Protocol.

`com.sun.identity.agents.config.amsso.cache.enable`

When enabled, the agent exposes SSO Cache through the agent SDK APIs.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > SSO > SSO Cache Enable.

`com.sun.identity.agents.config.attribute.cookie.encode`

When enabled, attribute values are URL encoded before being set as a cookie.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Attribute Cookie Encode.

`com.sun.identity.agents.config.attribute.cookie.separator`

Specifies the separator for multiple values of the same attribute when it is set as a cookie. Default: `|`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Cookie Separator Character.

`com.sun.identity.agents.config.attribute.date.format`

Specifies the `java.text.SimpleDateFormat` of date attribute values used when an attribute is set in an HTTP header. Default: `EEE, d MMM yyyy hh:mm:ss z`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Fetch Attribute Date Format.

`com.sun.identity.agents.config.audit.accesstype`

Types of messages to log based on user URL access attempts.

Valid values for the configuration file property include `LOG_NONE`, `LOG_ALLOW`, `LOG_DENY`, and `LOG_BOTH`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > Audit Access Types.

`com.sun.identity.agents.config.auth.handler`

Specifies custom authentication handler classes for users authenticated with the application server. The key is the web application name and the value is the authentication handler class name.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Custom Authentication Handler.

`com.sun.identity.agents.config.bypass.principal`

Specifies a list of principals the agent bypasses for authentication and search purposes, such as `guest` or `testuser`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Miscellaneous > Bypass Principal List.

`com.sun.identity.agents.config.cdsso.cdcservlet.url`

List of URLs of the available CDSSO controllers that the agent can use for CDSSO processing. For example, `http://openam.example.com:8080/openam/cdcservlet`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > SSO > CDSSO Servlet URL.

`com.sun.identity.agents.config.cdsso.clock.skew`

When set to a value other than zero, specifies the clock skew in seconds that the agent accepts when determining the validity of the CDSSO authentication response assertion.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > CDSSO Clock Skew.

`com.sun.identity.agents.config.cdsso.domain`

List of domains, such as `.example.com`, in which cookies have to be set in CDSSO.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cross Domain SSO.

`com.sun.identity.agents.config.cdsso.enable`

Enables Cross Domain Single Sign On.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cross Domain SSO.

`com.sun.identity.agents.config.cdsso.redirect.uri`

Specifies a URI the agent uses to process CDSSO requests.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > CDSSO Redirect URI.

`com.sun.identity.agents.config.cdsso.secure.enable`

When enabled, the agent marks the SSO Token cookie as secure, thus the cookie is only transmitted over secure connections.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cross Domain SSO.

`com.sun.identity.agents.config.cdsso.trusted.id.provider`

Specifies the list of OpenAM servers or identity providers the agent trusts when evaluating CDC Liberty Responses.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > CDSSO Trusted ID Provider.

`com.sun.identity.agents.config.change.notification.enable`

Enable agent to receive notification messages from OpenAM server for configuration changes.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Profile.

`com.sun.identity.agents.config.client.hostname.header`

If the agent is behind a proxy or load balancer, then the agent can get client IP and host name values from the proxy or load balancer. For proxies and load balancer that support providing the client IP and host name in HTTP headers, you can use the following properties.

When multiple proxies are load balancers sit in the request path, the header values can include a comma-separated list of values with the first value representing the client, as in `client,next-proxy,first-proxy`.

HTTP header name that holds the hostname of the client.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > Client Hostname Header.

`com.sun.identity.agents.config.client.ip.header`

Similar to `com.sun.identity.agents.config.client.hostname.header`, HTTP header name that holds the IP address of the client.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > Client IP Address Header.

`com.sun.identity.agents.config.conditional.login.url`

To conditionally redirect users based on the incoming request URL, set this property.

This takes the incoming request domain to match, a vertical bar (`|`), and then a comma-separated list of URLs to which to redirect incoming users.

If the domain before the vertical bar matches an incoming request URL, then the policy agent uses the list of URLs to determine how to redirect the user-agent. If the global property FQDN Check (`com.sun.identity.agents.config.fqdn.check.enable`) is enabled for the policy agent, then the policy agent iterates through the list until it finds an appropriate redirect URL that matches the FQDN check. Otherwise, the policy agent redirects the user-agent to the first URL in the list.

Property: `com.sun.identity.agents.config.conditional.login.url`

Examples: `com.sun.identity.agents.config.conditional.login.url[0]= login.example.com|http://openam1.example.com/openam/UI/Login, http://openam2.example.com/openam/UI/Login`, `com.sun.identity.agents.config.conditional.login.url[1]= signin.example.com|http://openam3.example.com/openam/UI/Login, http://openam4.example.com/openam/UI/Login`

If CDSSO is enabled for the policy agent, then this property takes CDSSO Servlet URLs for its values (`com.sun.identity.agents.config.cdsso.cdcservlet.url`), rather than OpenAM login URLs.

CDSSO examples: `com.sun.identity.agents.config.conditional.login.url[0]= login.example.com|http://openam1.example.com/openam/cdcservlet, http://openam2.example.com/openam/cdcservlet`, `com.sun.identity.agents.config.conditional.login.url[1]= signin.example.com|http://openam3.example.com/openam/cdcservlet, http://openam4.example.com/openam/cdcservlet`

`com.sun.identity.agents.config.conditional.logout.url`

The values take the incoming request URL to match and a comma-separated list of URLs to which to redirect users logging out.

Example: `com.sun.identity.agents.config.conditional.logout.url[0]= logout.example.com|http://openam1.example.com/openam/UI/Logout, http://openam2.example.com/openam/UI/Logout`

`com.sun.identity.agents.config.cookie.reset.domain`

Specifies how names from `com.sun.identity.agents.config.cookie.reset.name` correspond to cookie domain values when the cookie is reset.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cookie Reset Domain Map.

`com.sun.identity.agents.config.cookie.reset.enable`

When enabled, agent resets cookies in the response before redirecting to authentication.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cookie Reset.

`com.sun.identity.agents.config.cookie.reset.name`

List of cookies to reset if `com.sun.identity.agents.config.cookie.reset.enable` is enabled.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cookie Reset Name List.

`com.sun.identity.agents.config.cookie.reset.path`

Specifies how names from the `com.sun.identity.agents.config.cookie.reset.name` correspond to cookie paths when the cookie is reset.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > SSO > Cookie Reset Path Map.

`com.sun.identity.agents.config.default.privileged.attribute`

Specifies the list of privileged attributes granted to all users with a valid OpenAM session, such as `AUTHENTICATED_USERS`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Default Privileged Attribute.

`com.sun.identity.agents.config.filter.mode`

Specifies how the agent filters requests to protected web applications. The global value functions as a default, and applies for protected applications that do not have their own filter settings. Valid settings include the following.

`ALL`

Enforce both the J2EE policy defined for the web container where the protected application runs, and also OpenAM policies.

When setting the filter mode to `ALL`, set the Map Key, but do not set any Corresponding Map Value.

`J2EE_POLICY`

Enforce only the J2EE policy defined for the web container where the protected application runs.

`NONE`

Do not enforce policies to protect resources. In other words, turn off access management. Not for use in production.

`SSO_ONLY`

Enforce only authentication, not policies.

`URL_POLICY`

Enforce only OpenAM, URL resource based policies.

When setting the filter mode to `URL_POLICY`, set the Map Key to the application name and the Corresponding Map Value to `URL_POLICY`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > Agent Filter Mode.

`com.sun.identity.agents.config.fqdn.check.enable`

Enables checking of FQDN default value and FQDN map values.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > FQDN Check.

`com.sun.identity.agents.config.fqdn.default`

Fully qualified domain name that the users should use in order to access resources.

This property ensures that when users access protected resources on the web server without specifying the FQDN, the agent can redirect the users to URLs containing the correct FQDN.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > FQDN Default.

`com.sun.identity.agents.config.fqdn.mapping`

Enables virtual hosts, partial hostname and IP address to access protected resources. Maps invalid or virtual name keys to valid FQDN values so the agent can properly redirect users and the agents receive cookies belonging to the domain.

To map `myserver` to `myserver.mydomain.example`, enter `myserver` in the Map Key field, and enter `myserver.mydomain.example` in the Corresponding Map

Value field. This corresponds to `com.sun.identity.agents.config.fqdn.mapping[myserver]= myserver.mydomain.example`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > FQDN Virtual Host Map.

`com.sun.identity.agents.config.httpsession.binding`

When enabled the agent invalidates the HTTP session upon login failure, when the user has no SSO session, or when the principal user name does not match the SSO user name.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > HTTP Session Binding.

`com.sun.identity.agents.config.ignore.path.info`

When enabled, strip path info from the request URL while doing the Not Enforced List check, and URL policy evaluation. This is designed to prevent a user from accessing a URI by appending the matching pattern in the policy or not enforced list.

For example, if the not enforced list includes `/*.gif`, then stripping path info from the request URL prevents access to `http://host/index.html` by using `http://host/index.html?hack.gif`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Ignore Path Info in Request URL.

`com.sun.identity.agents.config.jboss.webauth.available`

When enabled, allow programmatic authentication with the JBoss container using the WebAuthentication feature. This feature works only with JBoss 4.2.2 to 7 when the `J2EE_POLICY` or `ALL` filter mode is in use.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > JBoss Application Server.

`com.sun.identity.agents.config.legacy.redirect.uri`

Specifies a URI the agent uses to redirect legacy user agent requests.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Legacy User Agent Redirect URI.

`com.sun.identity.agents.config.legacy.support.enable`

When enabled, provide support for legacy browsers.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Legacy User Agent Support Enable.

`com.sun.identity.agents.config.legacy.user.agent`

List of header values that identify legacy browsers. Entries can use the wildcard character, *.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Legacy User Agent List.

`com.sun.identity.agents.config.load.interval`

Interval in seconds to fetch agent configuration from OpenAM. Used if notifications are disabled. Default: 0

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Configuration Reload Interval.

`com.sun.identity.agents.config.local.log.rotate`

When enabled, audit log files are rotated when reaching the specified size.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Rotate Local Audit Log.

`com.sun.identity.agents.config.local.log.size`

Beyond this size limit in bytes the agent rotates the local audit log file if rotation is enabled. Default: 50 MB

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Local Audit Log Rotation Size.

`com.sun.identity.agents.config.locale.country`

The default country for the agent.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Locale Country.

`com.sun.identity.agents.config.locale.language`

The default language for the agent.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Locale Language.

`com.sun.identity.agents.config.log.disposition`

Specifies where audit messages are logged. By default, audit messages are logged remotely.

Valid values for the configuration file property include `REMOTE`, `LOCAL`, and `ALL`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Audit Log Location.

`com.sun.identity.agents.config.login.attempt.limit`

When set to a value other than zero, this defines the maximum number of failed login attempts allowed during a single browser session, after which the agent blocks requests from the user.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Login Attempt Limit.

`com.sun.identity.agents.config.login.content.file`

Full path name to the file containing custom login content when Use Internal Login is enabled.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Login Content File Name.

`com.sun.identity.agents.config.login.error.uri`

Specifies the list of absolute URIs corresponding to a protected application's `web.xml` `form-error-page` element, such as `/myApp/jsp/error.jsp`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Login Error URI.

`com.sun.identity.agents.config.login.form`

Specifies the list of absolute URIs corresponding to a protected application's `web.xml` `form-login-page` element, such as `/myApp/jsp/login.jsp`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Login Form URI.

`com.sun.identity.agents.config.login.url.prioritized`

When enabled, OpenAM uses the priority defined in the OpenAM Login URL list as the priority for Login and CDSSO URLs when handling failover.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Login URL Prioritized.

`com.sun.identity.agents.config.login.url.probe.enabled`

When enabled, OpenAM checks the availability of OpenAM Login URLs before redirecting to them.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Login URL Probe.

`com.sun.identity.agents.config.login.url.probe.timeout`

Timeout period in milliseconds for OpenAM to determine whether to failover between Login URLs when Login URL Probe is enabled. Default: 2000

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Login URL Probe Timeout.

`com.sun.identity.agents.config.login.url`

OpenAM login page URL, such as `http://openam.example.com:8080/openam/UI/Login`, to which the agent redirects incoming users without sufficient credentials so then can authenticate.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > OpenAM Login URL.

`com.sun.identity.agents.config.login.use.internal`

When enabled, the agent uses the internal default content file for the login.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Use Internal Login.

`com.sun.identity.agents.config.logout.application.handler`

Specifies how logout handlers map to specific applications. The key is the web application name. The value is the logout handler class.

To set a global logout handler for applications without other logout handlers defined, leave the key empty and set the value to the global logout handler class name, `GlobalApplicationLogoutHandler`.

To set a logout handler for a specific application, set the key to the name of the application, and the value to the logout handler class name.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Application Logout Handler.

`com.sun.identity.agents.config.logout.entry.uri`

Specifies the URIs to return after successful logout and subsequent authentication. The key is the web application name. The value is the URI to return.

To set a global logout entry URI for applications without other logout entry URIs defined, leave the key empty and set the value to the global logout entry URI, `/welcome.html`.

To set a logout entry URI for a specific application, set the key to the name of the application, and the value to the application logout entry URI, such as `/myApp/welcome.html`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Logout Entry URI.

`com.sun.identity.agents.config.logout.handler`

Specifies custom logout handler classes to log users out of the application server. The key is the web application name and the value is the logout handler class name.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Custom Logout Handler.

`com.sun.identity.agents.config.logout.introspect.enabled`

When enabled, the agent checks the HTTP request body to locate the Logout Request Parameter you set.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Logout Introspect Enabled.

`com.sun.identity.agents.config.logout.request.param`

Specifies parameters in the HTTP request that indicate logout events. The key is the web application name. The value is the logout request parameter.

To set a global logout request parameter for applications without other logout request parameters defined, leave the key empty and set the value to the global logout request parameter, `logoutparam`.

To set a logout request parameter for a specific application, set the key to the name of the application, and the value to the application logout request parameter, such as `logoutparam`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Logout Request Parameter.

`com.sun.identity.agents.config.logout.uri`

Specifies request URIs that indicate logout events. The key is the web application name. The value is the application logout URI.

To set a global logout URI for applications without other logout URIs defined, leave the key empty and set the value to the global logout URI, `/logout.jsp`.

To set a logout URI for a specific application, set the key to the name of the application, and the value to the application logout page.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Application Logout URI.

`com.sun.identity.agents.config.logout.url.prioritized`

When enabled, OpenAM uses the priority defined in the OpenAM Logout URL list as the priority for Logout URLs when handling failover.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Logout URL Prioritized.

`com.sun.identity.agents.config.logout.url.probe.enabled`

When enabled, OpenAM checks the availability of OpenAM Logout URLs before redirecting to them.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > Logout URL Probe.

`com.sun.identity.agents.config.logout.url.probe.timeout`

Timeout period in milliseconds for OpenAM to determine whether to failover between Logout URLs when Logout URL Probe is enabled. Default: 2000

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > Logout URL Probe Timeout.

`com.sun.identity.agents.config.logout.url`

OpenAM logout page URLs, such as `http://openam.example.com:8080/openam/UI/Logout`. The user is logged out of the OpenAM session when accessing these URLs.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > OpenAM Logout URL.

`com.sun.identity.agents.config.notenforced.ip.cache.enable`

When enabled, the agent caches evaluation of the not enforced IP list.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > Application > Not Enforced IP Cache Flag.

`com.sun.identity.agents.config.notenforced.ip.cache.size`

When caching is enabled, this limits the number of not enforced addresses cached. Default: 1000

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > Application > Not Enforced IP Cache Size.

`com.sun.identity.agents.config.notenforced.ip.invert`

Only enforce the not enforced list of IP addresses. In other words, enforce policy only for those client addresses and patterns specified in the list.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > Application > Not Enforced IP Invert List.

`com.sun.identity.agents.config.notenforced.ip`

No authentication and authorization are required for the requests coming from these client IP addresses.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Not Enforced Client IP List.

`com.sun.identity.agents.config.notenforced.refresh.session.idletime`

When enabled, the agent reset the session idle time when granting access to a not enforced URI, prolonging the time before the user must authenticate again.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Refresh Session Idle Time.

`com.sun.identity.agents.config.notenforced.uri.cache.enable`

When enabled, the agent caches evaluation of the not enforced URI list.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Not Enforced URIs Cache Enabled.

`com.sun.identity.agents.config.notenforced.uri.cache.size`

When caching is enabled, this limits the number of not enforced URIs cached.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Not Enforced URIs Cache Size.

`com.sun.identity.agents.config.notenforced.uri.invert`

Only enforce not enforced list of URIs. In other words, enforce policy only for those URIs and patterns specified in the list.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Invert Not Enforced URIs.

`com.sun.identity.agents.config.notenforced.uri`

List of URIs for which no authentication is required, and the agent does not protect access. You can use wildcards to define a pattern for a URI.

The * wildcard matches all characters except question mark (`?`), cannot be escaped, and spans multiple levels in a URI. Multiple forward slashes do not match a single forward slash, so * matches `mult/iple/dirs`, yet `mult/*/dirs` does not match `mult/dirs`.

The `-*-` wildcard matches all characters except forward slash (`/`) or question mark (`?`), and cannot be escaped. As it does not match `/`, `-*-` does not span multiple levels in a URI.

OpenAM does not let you mix * and `-*-` in the same URI.

Examples include `/logout.html`, `/images/*`, `/css/-*-`, and `/*.jsp?locale=*`.

Trailing forward slashes are not recognized as part of a resource name. Therefore `/images//` and `/images` are equivalent.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Not Enforced URIs.

`com.sun.identity.agents.config.policy.advice.use.redirect`

When enabled, the remote policy client is configured to use HTTP-Redirect instead of HTTP-POST for composite advice.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > Policy Client Service.

`com.sun.identity.agents.config.policy.env.get.param`

Specifies the list of HTTP GET request parameters whose names and values the agents sets in the environment map for URL policy evaluation by the OpenAM server.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > URL Policy Env GET Parameters.

`com.sun.identity.agents.config.policy.env.jsession.param`

Specifies the list of HTTP session attributes whose names and values the agents sets in the environment map for URL policy evaluation by the OpenAM server.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > OpenAM Services > URL Policy Env jsession Parameters.

`com.sun.identity.agents.config.policy.env.post.param`

Specifies the list of HTTP POST request parameters whose names and values the agents sets in the environment map for URL policy evaluation by the OpenAM server.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > URL Policy Env POST Parameters.

`com.sun.identity.agents.config.port.check.enable`

When enabled, activate port checking, correcting requests on the wrong port.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Port Check Enable.

`com.sun.identity.agents.config.port.check.file`

Specifies the name of the file containing the content to handle requests on the wrong port when port checking is enabled.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Port Check File.

`com.sun.identity.agents.config.port.check.setting`

Specifies which ports correspond to which protocols. The agent uses the map when handling requests with invalid port numbers during port checking.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Port Check Setting.

`com.sun.identity.agents.config.postdata.preserve.cache.entry.ttl`

POST data storage lifetime in milliseconds. Default: 300000.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > Post Data Preservation.

`com.sun.identity.agents.config.postdata.preserve.cache.noentry.url`

Specifies a list of application-specific URIs if the referenced Post Data Preservation entry cannot be found in the local cache because it has exceeded its POST entry TTL. Either the agent redirects to a URI in this list, or it shows an HTTP 403 Forbidden error.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Post Data Preservation.

`com.sun.identity.agents.config.postdata.preserve.enable`

Enables HTTP POST data preservation, storing POST data before redirecting the browser to the login screen, and then autosubmitting the same POST after successful authentication to the original URL.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Post Data Preservation.

`com.sun.identity.agents.config.postdata.preserve.stickysession.mode`

Specifies whether to create a cookie, or to append a query string to the URL to assist with sticky load balancing.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Post Data Preservation.

`com.sun.identity.agents.config.postdata.preserve.stickysession.value`

Specifies the key-value pair for stickysession mode. For example, a setting of `lb=myserver` either sets an `lb` cookie with `myserver` value, or adds `lb=myserver` to the URL query string.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Post Data Preservation.

`com.sun.identity.agents.config.privileged.attribute.mapping.enable`

When enabled, lets you use Privileged Attribute Mapping.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Enable Privileged Attribute Mapping.

`com.sun.identity.agents.config.privileged.attribute.mapping`

Maps OpenAM UUIDs to principal names specified in your web application's deployment descriptor, such as `com.sun.identity.agents.config.privileged.attribute.mapping [id\=manager,ou\=group,o\=openam] = am_manager_role` or `com.sun.identity.agents.config.privileged.attribute.mapping [id\=employee,ou\=group,o\=openam] = am_employee_role`

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > 0Privileged Attribute Mapping.

`com.sun.identity.agents.config.privileged.attribute.tolowercase`

Specifies how privileged attribute types should be converted to lower case.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Privileged Attributes To Lower Case.

`com.sun.identity.agents.config.privileged.attribute.type`

Specifies the list of privileged attribute types fetched for each user.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Privileged Attribute Type.

`com.sun.identity.agents.config.privileged.session.attribute`

Specifies the list of session property names, such as `UserToken` which hold privileged attributes for authenticated users.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Privileged Session Attribute.

`com.sun.identity.agents.config.profile.attribute.fetch.mode`

When set to `HTTP_COOKIE` or `HTTP_HEADER`, profile attributes are introduced into the cookie or the headers, respectively.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Profile Attribute Fetch Mode.

`com.sun.identity.agents.config.profile.attribute.mapping`

Maps the profile attributes to HTTP headers for the currently authenticated user. Map Keys are LDAP attribute names, and Map Values are HTTP header names.

To populate the value of profile attribute CN under `CUSTOM-Common-Name`: enter CN in the Map Key field, and enter `CUSTOM-Common-Name` in the Corresponding

Map Value field. This corresponds to `com.sun.identity.agents.config.profile.attribute.mapping[cn]=CUSTOM-Common-Name`.

In most cases, in a destination application where an HTTP header name shows up as a request header, it is prefixed by `HTTP_`, lower case letters become upper case, and hyphens (-) become underscores (_). For example, `common-name` becomes `HTTP_COMMON_NAME`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Profile Attribute Mapping.

`com.sun.identity.agents.config.redirect.attempt.limit`

When set to a value other than zero, this defines the maximum number of redirects allowed for a single browser session, after which the agent blocks the request.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Redirect Attempt Limit.

`com.sun.identity.agents.config.redirect.param`

Property used only when CDSSO is enabled. Only change the default value, `goto` when the login URL has a landing page specified such as, `com.sun.identity.agents.config.cdsso.cdcservlet.url = http://openam.example.com:8080/openam/cdcservlet?goto= http://www.example.com/landing.jsp`. The agent uses this parameter to append the original request URL to this cdcserlet URL. The landing page consumes this parameter to redirect to the original URL.

As an example, if you set this value to `goto2`, then the complete URL sent for authentication is `http://openam.example.com:8080/openam/cdcservlet?goto=http://www.example.com/landing.jsp?goto2=http://www.example.com/original.jsp`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Miscellaneous > Goto Parameter Name.

`com.sun.identity.agents.config.remote.logfile`

Name of file stored on OpenAM server that contains agent audit messages if log location is remote or all.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > Remote Log Filename.

`com.sun.identity.agents.config.repository.location`

Whether the agent's configuration is managed centrally through OpenAM (`centralized`) or locally in the policy agent configuration file (`local`).

Default: `centralized`

`com.sun.identity.agents.config.response.attribute.fetch.mode`

When set to `HTTP_COOKIE` or `HTTP_HEADER`, response attributes are introduced into the cookie or the headers, respectively. When set to `REQUEST_ATTRIBUTE`, response attributes are part of the HTTP response.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Response Attribute Fetch Mode.

`com.sun.identity.agents.config.response.attribute.mapping`

Maps the policy response attributes to HTTP headers for the currently authenticated user. The response attribute is the attribute in the policy response to be fetched.

To populate the value of response attribute `uid` under `CUSTOM-User-Name`: enter `uid` in the Map Key field, and enter `CUSTOM-User-Name` in the Corresponding Map Value field. This corresponds to `com.sun.identity.agents.config.response.attribute.mapping[uid]=Custom-User-Name`.

In most cases, in a destination application where an HTTP header name shows up as a request header, it is prefixed by `HTTP_`, lower case letters become upper case, and hyphens (`-`) become underscores (`_`). For example, `response-attr-one` becomes `HTTP_RESPONSE_ATTR_ONE`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Response Attribute Map.

`com.sun.identity.agents.config.response.header`

Specifies the custom headers the agent sets for the client. The key is the header name. The value is the header value.

For example, `com.sun.identity.agents.config.response.header[Cache-Control]=no-cache`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > Custom Response Header.

`com.sun.identity.agents.config.session.attribute.fetch.mode`

When set to `HTTP_COOKIE` or `HTTP_HEADER`, session attributes are introduced into the cookie or the headers, respectively. When set to `REQUEST_ATTRIBUTE`, session attributes are part of the HTTP response.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Session Attribute Fetch Mode.

`com.sun.identity.agents.config.session.attribute.mapping`

Maps session attributes to HTTP headers for the currently authenticated user. The session attribute is the attribute in the session to be fetched.

To populate the value of session attribute `UserToken` under `CUSTOM-userid`: enter `UserToken` in the Map Key field, and enter `CUSTOM-userid` in the Corresponding Map Value field. This corresponds to `com.sun.identity.agents.config.session.attribute.mapping[UserToken]=CUSTOM-userid`.

In most cases, in a destination application where an HTTP header name shows up as a request header, it is prefixed by `HTTP_`, lower case letters become upper case, and hyphens (`-`) become underscores (`_`). For example, `success-url` becomes `HTTP_SUCCESS_URL`.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Application > Session Attribute Map.

`com.sun.identity.agents.config.user.attribute.name`

Specifies the data store attribute that contains the user ID.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > User Attribute Name.

`com.sun.identity.agents.config.user.mapping.mode`

Specifies the mechanism used to determine the user ID.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Global > User Mapping Mode.

`com.sun.identity.agents.config.user.principal`

When enabled, OpenAM uses both the principal user name and also the user ID for authentication.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > User Principal Flag.

`com.sun.identity.agents.config.user.token`

Specifies the session property name for the authenticated user's ID. Default: `UserToken`.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Global > User Token Name.

`com.sun.identity.agents.config.verification.handler`

Specifies custom verification classes to validate user credentials with the local user repository. The key is the web application name and the value is the validation handler class name.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Application > Custom Verification Handler.

`com.sun.identity.agents.config.webservice.authenticator`

Specifies an implementation class of interface `com.sun.identity.agents.filter.IWebServiceAuthenticator` that can be used to authenticate web-service requests.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > Web Service Authenticator.

`com.sun.identity.agents.config.webservice.autherror.content`

Specifies a file the agent uses to generate an authorization error fault for the client application.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > Web Service Authorization Error Content File.

`com.sun.identity.agents.config.webservice.enable`

Enable web service processing.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > Advanced > Web Service Enable.

`com.sun.identity.agents.config.webservice.endpoint`

Specifies a list of web application end points that represent web services.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Web Service End Points.

`com.sun.identity.agents.config.webservice.internalerror.content`

Specifies a file the agent uses to generate an internal error fault for the client application.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Web Service Internal Error Content File.

`com.sun.identity.agents.config.webservice.process.get.enable`

When enabled, the agent processes HTTP GET requests for web service endpoints.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Web Service Process GET Enable.

`com.sun.identity.agents.config.webservice.responseprocessor`

Specifies a class implementing `com.sun.identity.agents.filter.IWebServiceResponseProcessor`, used to process web service reponses.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Web Service Response Processor.

`com.sun.identity.agents.config.xss.code.elements`

Specifies strings that, when found in the request, cause the agent to redirect the client to an error page.

For centralized configurations this property is configured under Access Control > *`Realm Name`* > Agents > J2EE > *`Agent Name`* > Advanced > Cross Site Scripting Detection.

`com.sun.identity.agents.config.xss.redirect.uri`

Maps applications to URIs of customized pages to which to redirect clients upon detection of XSS code elements.

For example, to redirect clients of MyApp to `/myapp/error.html`, use MyApp as the key and `/myapp/error.html` as the corresponding value.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > Advanced > Cross Site Scripting Detection.

`com.sun.identity.agents.notification.enabled`

When enabled, OpenAM sends notification about changes to policy.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > Enable Policy Notifications.

`com.sun.identity.agents.polling.interval`

Specifies the time in minutes after which the policy cache is refreshed. Default: 3

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > Policy Client Polling Interval.

`com.sun.identity.client.notification.url`

URL used by agent to register notification listeners.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > Global > Agent Notification URL.

`com.sun.identity.idm.remote.notification.enabled`

When enabled, receive notification from OpenAM to update user management data caches.

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > Enable Notification of User Data Caches.

`com.sun.identity.policy.client.booleanActionValues`

Specifies the values, such as `allow` and `deny`, that are associated with boolean policy decisions.

Default: `iPlanetAMWebAgentService|GET|allow|deny:iPlanetAMWebAgentService|POST|allow|deny`

For centralized configurations this property is configured under Access Control > `Realm Name` > Agents > J2EE > `Agent Name` > OpenAM Services > Policy Client Boolean Action Values.

`com.sun.identity.policy.client.cacheMode`

Set to cache mode `subtree` when only a small number of policy rules are defined. For large numbers of policy rules, set to `self`.

Default: `self`

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Policy Client Cache Mode.

`com.sun.identity.policy.client.clockSkew`

Time in seconds used adjust time difference between agent system and OpenAM. Clock skew in seconds = AgentTime - OpenAMServerTime.

Default: 10.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Policy Client Clock Skew.

`com.sun.identity.policy.client.resourceComparators`

Specifies the comparators used for service names in policy.

Default: `serviceType=iPlanetAMWebAgentService| class=com.sun.identity.policy.plugins.HttpURLResourceName|wildcard=*| delimiter=/| caseSensitive=false`

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Policy Client Resource Comparators.

`com.sun.identity.sm.cacheTime`

If notifications are not enabled and set to a value other than zero, specifies the time in minutes after which the agent polls to update cached service configuration data. Default: 1

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Service Data Cache Time.

`com.sun.identity.sm.notification.enabled`

When enabled, receive notification from OpenAM to update service configuration data caches.

For centralized configurations this property is configured under Access Control > *Realm Name* > Agents > J2EE > *Agent Name* > OpenAM Services > Enable Notification of Service Data Caches.