Oracle® Fusion Middleware

Administering Oracle Mobile Security Access Server

11*g* Release 2 (11.1.2.3)

E57098-01

April 2015

Oracle Fusion Middleware Administering Oracle Mobile Security Access Server, 11*g* Release 2 (11.1.2.3)

E57098-01

Contents

7 Configuring the SSL Keystore and Truststore

8 Managing Policies and Assertion Templates

9 Managing Log Files

10 Managing the MSAS Repository

Preface

This guide describes how to secure traffic between mobile devices and back-end URLs using Mobile Security Access Server. It describes how to create, manage, and configure logical MSAS instances, how to secure URLs using MSAS applications, and how to manage access policies and assertion templates and the MSAS repository.

Audience

This document is intended for system administrators using Mobile Security Access Server to secure traffic from mobile devices to back-end resources.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Mobile Security Suite and Oracle Identity Management documentation sets:

- *Release Notes for Oracle Identity Management*
- *Installing Oracle Mobile Security Access Server*
- *Installation Guide for Oracle Identity and Access Management*
- *Administering Oracle Mobile Security Suite*
- *Help Reference for Oracle Mobile Security Suite Consoles*
- *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*
- *WebLogic Scripting Tool Command Reference for Identity and Access Management*
- *High Availability Guide*
- *Administrator's Guide for Oracle Access Management*

- *Securing Applications with Oracle Platform Security Services*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with Mobile Security Access Server

This chapter describes how to get started using Mobile Security Access Server, including key concepts and an overview of the high-level architecture, a description of the administration tools, and a road map for using the product.

1.1 Understanding Mobile Security Access Server

Mobile Security Access Server (MSAS), a component in the Oracle Mobile Security Suite, provides a central access point for securing traffic from mobile devices to intranet resources.

Mobile Security Access Server:

- Can manage and secure both forward proxy and reverse proxy (virtual) web traffic.
- Provides single sign-on to back-end resources using various supported authentication mechanisms.
- Can be configured as an Oracle Access Manager (OAM) WebGate which is a web-server plug-in for Oracle Access Manager OAM that intercepts HTTP requests and protects resources by URL. For more information, see "Configuring an MSAS Instance as a WebGate" on page 6-68.

For an overview of the Oracle Mobile Security Suite, see "Understanding Oracle Mobile Security Suite" in *Administering Oracle Mobile Security Suite*.

1.2 Understanding Key Concepts in Mobile Security Access Server

The following topics introduce key concepts in Mobile Security Access Server.

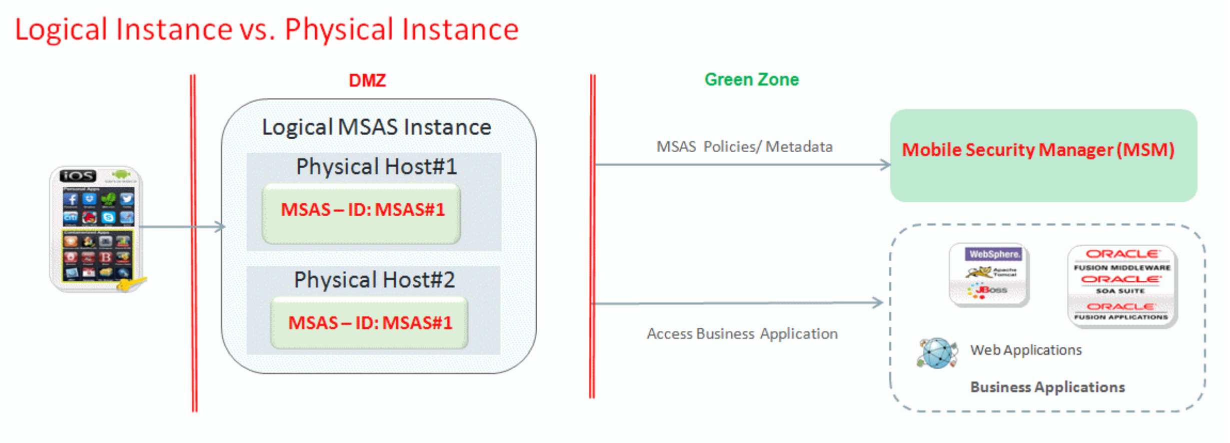
Logical and Physical MSAS Instances

MSAS supports the concept of logical and physical instances:

- A physical MSAS instance represents the actual physical machine on which MSAS is running and listening to requests, and provides security for traffic going through that physical instance. The physical MSAS instances can be running on different hosts, or on the same host on different ports. A typical environment will include multiple physical instances to provide high availability, distribute load, and improve performance. For details about configuring MSAS in a high availability environment, see "Configuring High Availability for Oracle Mobile Security Access Server" in *High Availability Guide*.
- A logical instance is an abstract concept to provide easy manageability of multiple physical instances that need to behave identically and have identical configuration. The MSM server hosts the definition of the logical MSAS instance so that the configuration common to all physical instances can be managed centrally (at the logical instance level) using the MSAS console or WLST commands.

Figure 1–1 illustrates the concept of logical and physical instances. In this figure, there is a logical instance named MSAS#1 running on two physical hosts, Physical Host#1 and Physical Host#2.

Figure 1–1 Relationship Between MSAS Physical and Logical Instances



MSAS Applications and URLs

MSAS applications are used to group related URLs to be proxied through the MSAS server. Each application contains the definition of one or more URLs (virtual or proxy) that you can secure using access policies and assertions. An MSAS application can be viewed as analogous to an EAR file, which is essentially a wrapper around modules and other files that can be deployed on an application server. An MSAS application serves as a wrapper of like URLs deployed on an MSAS instance, and can be exported and deployed on other MSAS instances.

The following types of applications are supported:

- Virtual applications—Applications defined in the MSAS environment that provide the ability to specify new URLs for back-end URLs. These new MSAS URLs are also called virtual URLs. You can then provide the MSAS URLs to clients and completely hide the back-end URLs. In this model, MSAS behaves as a reverse proxy for the back-end URL. For virtual URLS, access polices and assertions are attached at the HTTP method level.
- Proxy applications—Applications defined in the MSAS environment that specify back-end URLs that are proxied directly through the Mobile Security Access

Server. In this case, the Mobile Security Access Server acts as a forward proxy. The back-end URLs are visible to the client but the requests are proxied through the Mobile Security Access Server. For forward proxy URLs, access policies and assertions are attached at the URL-level.

- Blocked URL—Reserved application per MSAS instance that you can edit to specify URLs that are designated as inaccessible. This application is created by default when you create an MSAS instance.
- Direct URL—Reserved application per MSAS instance that you can edit to specify URLs that can be directly accessed and are not intercepted by the Mobile Security Access Server. This application is created by default when you create an MSAS instance.

MSAS applications are hosted on MSAS instances, and an instance can host multiple MSAS applications.

For details about MSAS applications, see Chapter 3, "Managing Mobile Security Access Server Applications."

Access Policies and Assertions

You secure the traffic between mobile devices and intranet resources, which you define as proxy and virtual URLs inside MSAS applications, using access policies and assertions. Mobile Security Access Server provides a set of predefined access policies and assertion templates that are installed with the MSAS Management Server. These predefined policies and assertions are based on common best-practice policy patterns in customer deployments, and can be used to configure both authentication and authorization.

The predefined policies and assertions can be attached at three policy enforcement points: on request, when invoking the back-end application, and on response. The configuration properties included with the predefined policies allow you to override certain configuration settings, such as the CSF key used for storing the signature-key password. The scope for the configuration property override value is limited to the specific policy attachment.

Oracle recommends that you do not edit these predefined policies and assertions. If you wish to edit a predefined policy or assertion template, Oracle recommends that you clone the artifact and then edit it.

You can manage these policies from the MSAS console. For details about managing these policies and assertion templates, see "Managing Policies and Assertion Templates" on page 8-1. For reference information, see *Policy and Assertion Template Reference for Mobile Security Access Server*.

Authentication Mechanisms

Mobile Security Access Server supports the following authentication mechanisms from mobile devices:

- Kerberos Password Authentication (KINIT)—Authentication mechanism for Kerberos 5 used to obtain and cache a Ticket Granting Ticket (TGT) for a principal. A TGT is a ticket that the Kerberos Key Distribution Center grants to a client after authenticating it. KINIT processing is delegated to internal modules in MSAS that invoke the KDC server and authenticate the user.
- Kerberos PKI-based Authentication (PKINIT) with initial registration using a time-limited passcode—Preauthentication mechanism for Kerberos 5 that uses X.509 certificates to authenticate the clients to the KDC. Similar to KINIT

authentication, PKINIT processing is delegated to internal modules in MSAS that invoke the KDC server and authenticate the user.

- OAuth Confidential Client—Client-server authentication mechanism in the Oracle Access Manager Mobile and Social (OAMMS) server where the client is an OAuth 2.0 client and the OAMMS server is an OAuth 2.0 implementation. In this type of authentication, the client is 2-legged. The confidential client can obtain JWT User Token (referred to as User Identity Assertion) using client id, secret, user id and password. The confidential client can obtain OAuth 2.0 Access Token using standard OAuth 2.0 JWT user assertion flow on behalf of the resource owner.
- OAuth Mobile Client—Client-server authentication model where the client is a mobile application and the server is Oracle Access Manager Mobile and Social (OAMMS). In this authentication model, a workspace is dynamically registered with the OAMMS server through MSAS and the workspace obtains the JWT Client Token after successful workspace registration. The registered workspace can perform authentication against OAMMS through MSAS and MSAS can obtain JWT User Token after the successful authentication process.

The URL of these authentication endpoints is provided in the configuration JSON files that the Mobile Security Workspace app presents to the user during registration. The type of configuration JSON file selected determines the Workspace app authentication mechanism.You can edit the configuration of these authentication endpoints as described in "Configuring Authentication Endpoints" on page 6-14.

1.3 Mobile Security Access Server Architecture Overview

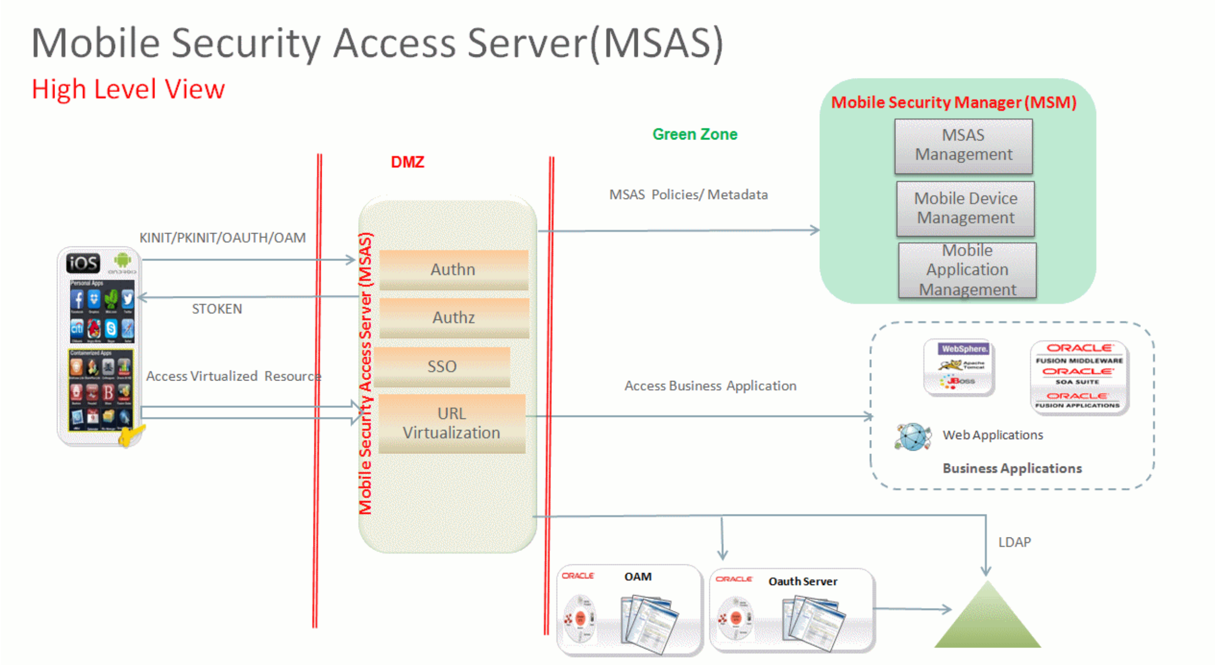
Mobile Security Access Server consists of the following components:

- Mobile Security Access Server (MSAS)—Run-time engine used for virtualizing and securing traffic. This component serves as the gateway between mobile devices and back-end resources. MSAS is hosted in its own server process, and runs in the demilitarized zone (DMZ), providing a first line of defense.

  The purpose of the DMZ is to add an additional layer of security to an organization’s local area network (LAN). Using a DMZ, an external entity has direct access only to components in the DMZ, rather than any other parts of the network.

  MSAS server handles the single sign on to the back-end URLs using various supported authentication mechanism and enforces the security policies on request from the client, when invoking enterprise applications, and on response to the client.
- Mobile Security Manager (MSM)—MSM contains a management component for MSAS that contains services used to manage MSAS. MSM is deployed on WebLogic Server, in the green zone.
- Repository—Persistent store (database) that consists of an MDS store for access policies and assertion templates, and MSAS configurations including MSAS applications and URLs.

Figure 1–2 Mobile Security Access Server Architecture Diagram



1.4 Mobile Security Access Server Administration Tools

System administrators can use the tools described in Table 1–1 to configure and manage MSAS instances.

Table 1–1 Mobile Security Access Server Configuration Tools

Tool	Description
configMSAS.sh	Configuration script installed with the MSAS run-time server. Use this script to create MSAS instances and register them with the MSAS Management Server.
idmConfigTool	Configuration script used to configure the identity store, keystore, and truststore used by MSAS. For MSAS configuration, this script must be used with the `-configOMSS mode=OMSAS` option.
MSAS console pages in the Oracle Access Management Console	Administration interface component of the Oracle Access Management console used for the administration of MSAS. The MSAS console pages, which are accessible from the Mobile Security Launch Pad, are installed with the Mobile Security Manager (MSM). Use these console pages to: ■ Register logical MSAS instances with the Management Server. ■ Configure identity store profiles. ■ Configure message security, including associating Keystore Service (KSS) stripes with MSAS instances, choosing encryption/signature keys and certificates, and specifying message security settings such as clock skew, expiration, and so on. ■ Configure trusted issuers (SAML and JWT). ■ Configure run-time settings such as proxy server and outbound message settings. ■ Configure MSAS endpoints for initial authentication—KINIT, PKINIT, OAuth2 Mobile Client, and OAuth2 Confidential Client. ■ Manage access policies and assertion templates. ■ Manage virtual and proxy applications (create, view, delete, search). ■ Add reverse proxy URLs to virtual applications and secure them by attaching access policies and assertions, and configuring policy property overrides. ■ Add forward proxy URLs to proxy applications and secure them by attaching access policies and assertion templates, and configuring policy property overrides. ■ Configure role-based authorization. ■ Manage diagnostic and message logs.
WLST	Command-line scripting environment provided with the Mobile Security Manager. Use the MSAS WLST commands to: ■ Configure identity store profiles. ■ Configure message security, including associating Keystore Service (KSS) stripes with MSAS instances, choosing encryption/signature keys and certificates, and specifying message security settings such as clock skew, expiration, and so on. ■ Configure trusted issuers (SAML and JWT). ■ Configure run-time settings such as proxy server and outbound message settings. ■ Configure MSAS endpoints for initial authentication—KINIT, PKINIT, OAuth2, and OAMMS. ■ Migrate MSAS application metadata between environments. ■ Manage diagnostic and message log levels.

1.5 Roadmap for Installing and Using Mobile Security Access Server

Table 1–2 provides a roadmap for installing and using Mobile Security Access Server.

Table 1–2 Roadmap for Installing and Using Mobile Security Access Server

Task	More information
Review prerequisites for installing MSAS	"Prerequisites for Installing Mobile Security Access Server" in *Installing Oracle Mobile Security Access Server.*

Table 1–2 (Cont.) Roadmap for Installing and Using Mobile Security Access Server

Task	**More information**
Install MSAS	"Installing Mobile Security Access Server" in *Installing Oracle Mobile Security Access Server*
Create an MSAS instance using the `configMSAS` tool.	"Configuring an MSAS Instance" in *Installing Oracle Mobile Security Access Server*. **Note:** You can also create a logical instance using the MSAS console as described in "Creating and Registering a Logical MSAS Instance" on page 2-3. You can configure the instance using the console, then bind it to a physical instance using the `configMSAS` tool.
Configure identity store, keystore, and truststore for an MSAS instance using the `idmConfig` utility.	"Configuring the Identity Store and Keystores for the MSAS Instance" in *Installing Oracle Mobile Security Access Server*.
Start the MSAS server	"Starting and Stopping the MSAS Server" in *Installing Oracle Mobile Security Access Server*.
View MSAS instances in the MSAS console	"Viewing MSAS Instances in the Environment" on page 2-1.
Manage the security configuration of an MSAS instance, including identity store profiles, keystores, trusted issuers, SSL, authentication endpoints, and system settings.	Chapter 6, "Configuring a Mobile Security Access Server Instance."
Create proxy and virtual applications and define URLs.	■ "Managing Mobile Security Access Server Applications" on page 3-1 ■ "Configuring Single Sign-On (SSO) for OAM WebGate and Oracle WSM Protected Resources" on page 6-65 ■ "Configuring Single Sign-On for Kerberos and NTLM Protected Resources" on page 6-66
Secure URLs in virtual applications.	"Attaching Policies and Assertions to Virtual Applications" on page 5-2.
Secure URLs in proxy applications.	"Attaching Policies and Assertions to Proxy Applications" on page 5-4.
Configure web settings, such as direct and blocked URLs.	Chapter 4, "Configuring Web Settings in MSAS."
Configure role-based authorization	"Configuring Authorization in MSAS Applications" on page 5-9.
Configure SSL keystores and truststores	Chapter 7, "Configuring the SSL Keystore and Truststore".
Manage the repository	Chapter 10, "Managing the MSAS Repository"

2 Managing Mobile Security Access Server Instances

You manage Mobile Security Access Server (MSAS) instances using the Environments page in the MSAS console component of the OAM console. From this page you can view all the MSAS instances in the environment, view the configuration for each instance, and create new logical MSAS instances.

2.1 Viewing MSAS Instances in the Environment

To view all the MSAS instances in the environment:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, select **Environments** in the Mobile Security Access Server section.

   The Environments page opens in a new tab. From this page you can:

   - View the total number of MSAS instances configured in the environment.
   - View the total number of MSAS applications deployed on the MSAS instances. For more information about MSAS applications, see Chapter 3, "Managing Mobile Security Access Server Applications."
   - View the total number of URLs configured in all the applications. For more information about MSAS application URLs, see "Managing URLs in an MSAS Application" on page 3-9.
   - Create a new logical MSAS instance and register it in the environment. For instructions, see "Creating and Registering a Logical MSAS Instance" on page 2-3.
3. Click **MSAS** or **Instances** in the MSAS tile.

The MSAS Instances Summary page opens in a new tab. You can use this page to view summary information for each instance.

The first 8 instance in the environment are displayed. Click **Show More** to show additional instances.

4. Use the **Search** field to refine the list of instances or to locate a specific instance. Enter all or part of a name in the **Search** field and press the search icon.

2.2 Viewing the Configuration of an MSAS Instance

To view the configuration of an MSAS instance:

1. Navigate to the MSAS Instances Summary page as described in "Viewing MSAS Instances in the Environment" on page 2-1.
2. If necessary, use the **Search** field to refine the list of instances or to locate a specific instance. Enter all or part of a name in the **Search** field and press the search icon.
3. Click **Configure** or the instance name in the tile for the instance to be viewed.

   The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance. The General tab on this page provides summary information about the instance, such as Name, Display Name, MSAS URL of the physical instance, instance statistics such as number of applications and URLs in the instance, and version information.
4. Click the remaining tabs on this page to view the configuration details. Additional information and procedures for configuring the instance are provided in Chapter 6, "Configuring a Mobile Security Access Server Instance."

2.3 Synchronizing MSAS Instance Configuration

When you change an MSAS instance, for example to modify the configuration or edit applications or URLs, synchronization with the run-time server typically occurs at a user-specified polling interval specified using a cache refresh property. As a result, the changes may not go into effect until the next scheduled polling interval.

> **Note:** The default polling interval is 86,400,000ms (24 hours). You can adjust this property setting as described in "Configuring the Cache Refresh Time" on page 6-13 and "Configuring the Cache Refresh Time Using WLST" on page 6-49.

You can force immediate synchronization of the changes as follows:

1. Navigate to the MSAS Instances Summary page as described in "Viewing MSAS Instances in the Environment" on page 2-1.
2. If necessary, use the **Search** field to refine the list of instances or to locate the specific instance for which you want to synchronize the changes. You can also enter all or part of a the instance name in the **Search** field and press the search icon.
3. Click **Synchronize** in the tile for the instance that contains the changes to be synchronized.

   An information message displays indicating that the synchronization process has been initiated for the instance.

Note: It may take approximately 60 seconds for the changes to go into effect in the run time.

4. Click **OK** in the message window.

2.4 Creating and Registering a Logical MSAS Instance

You can create a logical MSAS instance and register it with the Mobile Security Manager (MSM) using the MSAS console. You can use this logical instance to create applications, and configure the instance, without being bound to a physical machine. This might be useful in a test to production environment, for example, where you can automate the data creation process without being dependent on the availability of the physical hardware.

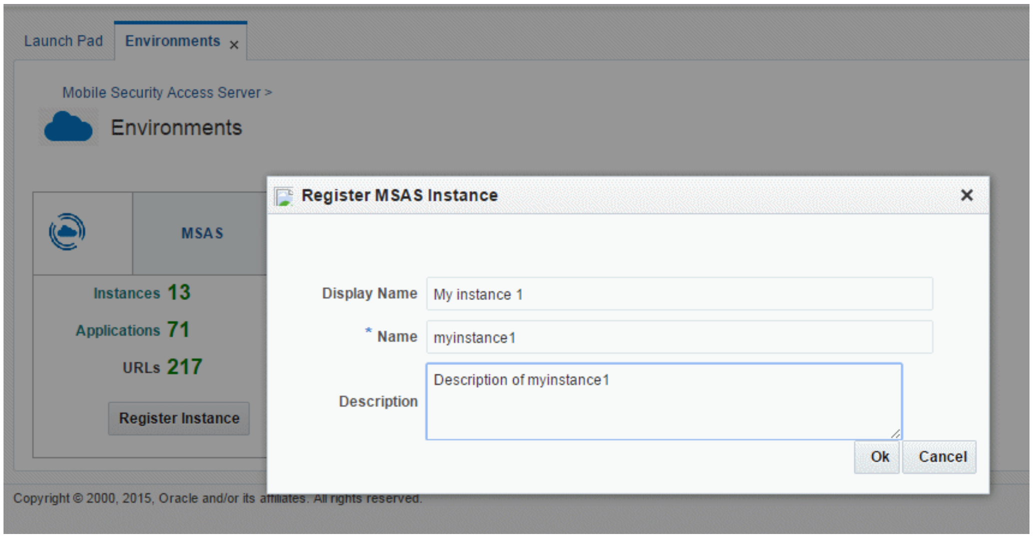
Once the configuration is complete, you can bind this logical instance to a physical instance using the `configMSAS` configuration script. For more information, see "Binding a Logical MSAS Instance to a Physical Instance" in *Installing Oracle Mobile Security Access Server.*

To create a logical MSAS instance and register it with the MSM server:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, select **Environments** in the Mobile Security Access Server tile.

   The Environment Summary page opens in a new tab.
3. Click **Register Instance**.
4. In the Register MSAS Instance window, complete the fields then click **OK**. The Register MSAS Instance window is shown in Figure 2–1.

Figure 2–1 Register MSAS Instance Window



Display Name: My instance 1

* Name: myinstance1

Description: Description of myinstance1

Field	Description
Display Name	Optionally, enter a meaningful name that can be used to identify the instance in the console.
Name	Enter a name for the MSAS instance. The name must: ■ Be unique within the MSAS environment. ■ Adhere to the XML xs:NCName format using only valid NCName ASCII characters. For example, it must start with a letter or underscore (_), and cannot contain any space characters or colons (:). For the NCName format definition, see the *W3C document Namespaces in XML 1.0 (Third Edition)* at `http://www.w3.org/TR/REC-xml-names/#NT-NCName`
Description	Optionally, enter a short description of the MSAS instance.

5. Configure the instance using the MSAS Instance Configuration page. This page displays in a new tab using the instance ID as the tab name. Use the subtabs on this page to configure the instance. Much of the configuration can also be completed using WLST commands. For details, see Chapter 6, "Configuring a Mobile Security Access Server Instance."

2.5 Changing the MSM Server Associated with an MSAS Instance

To change the Mobile Security Manager URL to which an MSAS instance is registered, you need to run the MSAS configuration script (`configMSAS`) on the machine on which you configured the MSAS instance. You can perform this update in interactive mode by responding to the prompts, or in silent mode using a properties file.

1. If a logical MSAS instance with the same name does not already exist on the MSM server to which you are assigning this instance, log into the OAM console on that MSM server to access the MSAS console pages and create a logical instance using the same name as the instance to be updated. For details about creating a logical MSAS instance, see "Creating and Registering a Logical MSAS Instance" on page 2-3.

2. Go to the machine on which the MSAS instance to be updated is configured and change to the *`ORACLE_HOME`*`/omsas/bin` directory, where *`ORACLE_HOME`* is the directory you specified for Oracle Home when you installed Mobile Security Access Server, for example, `/u01/oracle/omsas/Oracle_MSAS`.

3. Enter the following command to start the MSAS configuration script in update mode:

```
sh configMSAS.sh -update
```

Note: To execute this script in silent mode, provide the name of the properties file on the command line, for example:

```
sh configMSAS.sh -update myupdatefile.properties
```

You must include the following properties in the properties file:

- `MSM_URL`
- `MSM_USER_NAME`
- `MSM_PASS`
- `MSAS_INSTANCE_ID`
- `MSAS__INSTANCE_ROOT_DIR`

For descriptions of these properties, and details about executing this script in silent mode, see "Using Silent Mode to Configure an MSAS Instance" in *Installing Oracle Mobile Security Access Server*.

4. Respond to each prompt as described in Table 2–1. If you make an error and need to exit the script without completing the update, press `Ctrl-c`.

Table 2–1 MSAS Configuration Script Prompts to Update Bootstrap Credentials

Prompt	Description
Enter the MSAS Instance ID	Enter the name of the MSAS instance for which you want to update the bootstrap credentials.
Enter the MSAS Instance Root Dir	Enter the full path to the directory containing the MSAS instance to be updated.
Do you want to update the Mobile Security Manager (MSM) URL	Enter `y` to update the Mobile Security Manager URL or `n` to use the existing URL.
Enter the Mobile Security Manager (MSM) URL	If you entered `y` to update the MSM URL, enter the URL for the MSM Server to which you want this MSAS instance to be registered. For an SSL URL, use: `https://`*`host_name`*`:`*`port_number`* For a non-SSL URL, use the following format: `http://`*`host_name`*`:`*`port_number`* In both formats, *`host_name`* represents the host name or IP address of the host machine, and *`port_number`* represents the listen port for the MSM server. If you have only one MSM Server, this is typically port 14180 (non-SSL) or 14181 (SSL). If you configured other ports when you created the MSM domain, be sure to enter the appropriate MSM server listen port.
Do you want to update the Mobile Security Manager (MSM) credential	Enter `y` to update the MSM credential, or `n` to keep the existing credential. If you are updating the MSM URL, be sure to enter the correct credentials for the new MSM server.
Enter the Username to connect to the Mobile Security Manager (MSM)	Enter the username for the updated credential.

Table 2–1 (Cont.) MSAS Configuration Script Prompts to Update Bootstrap Credentials

Prompt	Description
Enter the Password to connect to the Mobile Security Manager (MSM)	Enter the password for the new credential.

When the process completes successfully, the following message displays:

```
The Instance For MSAS Instance Id - instance_name Configured Successfully.
```

You will now be able to manage this MSAS instance using the MSAS console or WLST commands on the MSM server with which this instance is now registered.

> **Note:** The list of MSAS URLs on the **General** tab of the MSAS Instance Configuration page in the MSAS console is not updated to reflect the physical instances to which this logical instance is linked.

2.6 Deleting a Logical MSAS Instance

To delete a logical MSAS instance:

1. Navigate to the MSAS Instances Summary page as described in "Viewing MSAS Instances in the Environment" on page 2-1.
2. If necessary, use the **Search** field to refine the list of instances or to locate a specific instance. Enter all or part of a name in the **Search** field and press the search icon.
3. Delete the desired instance by clicking the **X** in the upper-right corner of the instance tile.

   You are prompted to confirm that you want to delete the instance. Click **OK**. Once deleted, the instance is deleted from the MSAS Instances Summary page and is no longer managed by the MSM server.

> **Note:** When the logical instance ID is deleted in the console, any associated physical instances can not be managed from MSM and will result in runtime errors.

2.7 Starting and Stopping MSAS

To start or stop MSAS:

1. Change to the `MW_HOME/instances/instance_name/bin` directory, where *`MW_HOME`* is the Middleware home directory in which you installed Mobile Security Access Server and *`instance_name`* is the name of the MSAS instance you want to start or stop.
2. To start MSAS, enter the following command:

   ```
   sh startServer.sh
   ```

3. To stop MSAS, enter the following command:

   ```
   sh stopServer.sh
   ```

3 Managing Mobile Security Access Server Applications

Mobile Security Access Server (MSAS) applications group related URLs to be proxied through the MSAS server. You can secure these URLs using access policies and assertions. Each application contains the definition of the URLS to be proxied, as well as the security artifacts and policies attached to each URL.

This chapter includes the following sections:

3.1 Mobile Security Access Server Application Types

Mobile Security Access Server (MSAS) can act as a forward proxy or a reverse proxy for URL traffic.

- As a forward proxy, MSAS acts as an intermediary allowing clients to directly access back-end resources.
- As a reverse proxy, MSAS hides the back-end resources from the clients. The response to the client looks like it originated from MSAS.

MSAS uses the concept of application types to capture how MSAS handles a particular URL, that is as a forward proxy or a reverse proxy. Each application:

- Contains the definition of one or more virtual URLs or proxy URLs.
- Contains related security artifacts and access policies attached to each URL.

The applications are deployed to MSAS instances, and can be exported and imported from test to production environments.

Mobile Security Access Server supports the types of applications described in the following table.

Type	Console Icon	Description
Virtual Application	☁	Virtual applications allow customers to create new URLs in MSAS for back-end URLs. These new MSAS URLs are also called virtual URLs. Customers can then provide the MSAS URLs to clients and completely hide the back-end URLs. In this model, MSAS behaves as a reverse proxy for the back-end URL.
Proxy Application	☁	Applications defined in the MSAS environment that specify back-end URLs that will be proxied directly through the Mobile Security Access Server. In this case, the Mobile Security Access Server acts as a forward proxy. The back-end URLs are visible to the client but the requests are proxied through the Mobile Security Access Server.
Direct URL	Direct	Reserved application per MSAS instance that you can edit to specify URLs that are directly accessed and are not intercepted by the Mobile Security Access Server. This application is created by default when you create an MSAS instance. You cannot create or delete this type of application. For details about using this application to configure directly-accessed URLs, see "Configuring Direct URLs."
Blocked URL	Block	Reserved application per MSAS instance that you can edit to specify URLs that are designated as inaccessible. This application is created by default when you create an MSAS instance. You cannot create or delete this type of application. For details about using this application to configure blocked URLs, see "Configuring Blocked URLs."

3.2 Reserved Applications in MSAS

Table 3–1 describes the reserved applications that are created by default when you create an MSAS instance.

Table 3–1 Default Virtual and Proxy Applications

Application Name	Type	Description
BLOCK	Blocked URL	Used to specify URLs that are designated as inaccessible, or blacklisted. You can add URLS to, and delete URLs, from this application, but you cannot edit the application name, or delete the application.
DIRECT	Direct URL	Used to specify URLs that can be directly accessed and are not intercepted by the Mobile Security Access Server. You can add and delete URLS, but you cannot edit the application name, or delete the application

Table 3–1 (Cont.) Default Virtual and Proxy Applications

Application Name	Type	Description
msm-reverse-proxy	Virtual	Provides virtual URLs for Mobile Security Manager (MSM) services such as Mobile Device Management (MDM) and Mobile Application Management (MAM). You can configure the URLs defined in this application, but you cannot add or delete URLs, change the name of the application, or delete the application. **WARNING:** Deleting this application will destabilize an MSAS instance and lead to undefined behavior. In this case you will need to create an entirely new instance.
msm	Proxy	Used to secure forward-proxy requests for MSM services such as Enhanced Client/Proxy (ECP) and Mobile File Manager (MFM). You can configure the URLs defined in this application, but you cannot add or delete URLs, change the name of the application, or delete the application. **WARNING:** Deleting this application will destabilize an MSAS instance and lead to undefined behavior. In this case you will need to create an entirely new instance.
Default URL	Proxy	This application is provided for convenience purposes and is used to secure all forward-proxy requests by default. It contains a wildcard path (/) that applies to all URLs in the environment that are not defined explicitly in a proxy application. By using a default URL, you do not need to define every URL in the system using a proxy URL. You can edit the URL defined in the application, and you can add or delete proxy URLs. You can edit the name of this application, and delete it.
OAM Pass-through Proxy App	Proxy	Proxy application that contains default pass-through OAM URLs required when using MSAS as a WebGate. **Warning:** If you are using MSAS as a WebGate, you should not delete this application. Doing so can lead to undefined behavior.

3.3 Viewing MSAS Applications in the Environment

To view all the applications in the MSAS environment:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Applications** in the Mobile Security Access Server section to display the MSAS Applications page.

   Alternatively, from the Mobile Security Launch Pad, click **Environments**, then click **Applications** in the MSAS tile.

   The MSAS Applications page is displayed.
3. Click **Load More Items** to display additional applications as required.

3.4 Creating a Virtual Application

Virtual MSAS applications include one or more virtual URLs, or reverse-proxy URLs. In reverse-proxy, you create a virtual URL to hide the actual URL from the client.

To create a virtual application:

1. Navigate to the MSAS Applications page:
   - From the Mobile Security Launch Pad in the OAM console, click **Applications** in the Mobile Security Access Server section.
   - Alternatively, from the Mobile Security Launch Pad, click **Environments**, then click **Applications** in the MSAS tile.
2. Click **+Create**, then **Virtual Application**.
3. In the Create Virtual Application window, provide the name, display name, description, and MSAS instance for the application and click **Save**.

Field	Description
Name	Enter a name for the application. The name must: ■ Be unique within the MSAS instance. ■ Adhere to the XML xs:NCName format using only valid NCName ASCII characters. For example, it must start with a letter or underscore (_), and cannot contain any space characters or colons (`:`). For the NCName format definition, see the *W3C document Namespaces in XML 1.0 (Third Edition)* at `http://www.w3.org/TR/REC-xml-names/#NT-NCName`
Display Name	Optionally, enter a meaningful name that can be used to identify the application in the console.
Description	Optionally, enter a brief description of the application.
MSAS Instance	From the menu, select the instance that will contain the application.

4. In the URLs page, click **+URL** or **Create URL** to add virtual URLs to the application.
5. In the Add URL window, click **Add** to configure the virtual URL and enter the host URL, name, MSAS URI, and HTTP method. To add more than one URL, click **Add** again and complete the fields. When you have defined the URLs, click **Save**. Note that once you click **Save**, you need to click **+URL** again to add more URLs.

Field	Description
Host URL	Enter the URL to add to the application. This URL will not be visible to clients. For example, `http://host1:port1/actualURL`.
Name	Enter a meaningful name for the virtual URL. This name will be used to identify the virtual URL in the console.
MSAS URI	Enter the virtual URL that will be visible to clients. It must be unique within the MSAS instance and will identify the relative path of the virtualized URL. For example `/actualURL`. The full virtual web application URL will resolve to `http://msas_host:msas_port/actualURL`.

Field	Description
HTTP Method	Select the HTTP method to use for the virtual URL. ■ GET—Retrieves the information specified in the request URI. ■ POST—Requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request URI. ■ PUT—Requests that the target resource be created or modified with the entity enclosed in the request message. ■ HEAD— Identical to GET but without the message body in the response. ■ OPTIONS—Returns the HTTP methods that the server supports for the URL. ■ TRACE—Loops the received request back to the client so that they can see what was received by the server and any intermediaries. ■ CONNECT—Not supported. ■ DELETE—Delete the resource specified in the request URL. ■ All—All HTTP verbs. For details about HTTP methods, see the *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content* RFC document at `https://tools.ietf.org/html/rfc7231`.
Description	Optionally, enter a description for the virtual URL.
2-way SSL	Reserved for future use.

6. On the URLs page, click the URL icon or the URL Name to open the URL Policy Configuration page for the URL. The URL Policy Configuration page opens in a new tab. From this page, you can secure the URL at the policy enforcement points. For details, see Chapter 5, "Securing Mobile Security Access Server Resources."

3.5 Creating a Proxy Application

Proxy applications include one or more forward proxy URLs. In forward proxy, the client is aware of the URL and can access it directly using a proxy server configured on the client side.

To create a proxy application:

1. Navigate to the MSAS Applications page:
   - From the Mobile Security Launch Pad in the OAM console, click **Applications** in the Mobile Security Access Server section.
   - Alternatively, from the Mobile Security Launch Pad, click **Environments**, then click **Applications** in the MSAS tile.
2. Click **+Create**, then **Proxy Application**.
3. In the Create Proxy Application window, provide the name, display name, description, and MSAS instance for the application and click **Save**.

Field	Description
Name	Enter a name for the application. The name must: - Be unique within the MSAS instance. - Adhere to the XML xs:NCName format using only valid NCName ASCII characters. For example, it must start with a letter or underscore (_), and cannot contain any space characters or colons (:). For the NCName format definition, see the *W3C document Namespaces in XML 1.0 (Third Edition)* at `http://www.w3.org/TR/REC-xml-names/#NT-NCName`
Display Name	Optionally, enter a meaningful name that can be used to identify the application in the console.
Description	Optionally, enter a brief description of the application.
MSAS Instance	From the menu, select the instance that will contain the application.

4. In the Proxy URLs page, click **+Proxy URL** or **Create URL** to add proxy URLs to the application.
5. In the Add Proxy URL window, click **Add** to configure the proxy URL and enter a host URL, and name. To add more than one URL, click **Add** again and complete the fields. When you have defined the URLs, click **Save**. Note that once you click **Save**, you need to click **+Proxy URL** again to add more URLs.

Field	Description
Host URL	Enter the URL to add to the application. It must be unique to all applications in the MSAS instance.
Name	Enter a meaningful name for the proxy URL. This name will be used to identify the proxy URL in the console.
Description	Optionally, enter a description for the proxy URL.
2-way SSL	Reserved for future use.

6. On the URLs page, click the URL icon or the URL Name to open the URL Policy Configuration page for the URL in a new tab. From this page, you can secure the URL at the policy enforcement points. For details, see Chapter 5, "Securing Mobile Security Access Server Resources."

3.6 Viewing the Applications in an MSAS Instance

To view the applications in an MSAS instance:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Applications** in the Mobile Security Access Server section to display the MSAS Applications page.
3. Use the **Search** field to filter the list of applications by MSAS Instance Name:
   a. From the Search menu, select **MSAS Instance Name**.
   b. Enter all or part of the MSAS Instance Name in the field and click the Search icon.

   An initial set of applications in the instance is displayed.

4. Click **Load More Items** to display additional applications as required.

Alternatively, you can navigate to the MSAS applications page directly from the instance. When you do so, only the applications in the instance are displayed. To do so:

1. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Environments** in the Mobile Security Access Server section.
3. Click **MSAS** or **Instances** in the MSAS tile to open the MSAS Instances Summary page.
4. If necessary, use the **Search** field to refine the list of instances or to locate a specific instance. Enter all or part of a name in the **Search** field and press the search icon.
5. Click **Applications** in the tile for the desired instance. The first 5 applications in the instance are displayed in the MSAS Applications page.
6. Click **Load More Items** to display additional applications as required.

3.7 Searching for MSAS Applications

To search for applications in the environment or in an instance:

1. Navigate to the MSAS Applications page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

2. Select the type of search and enter the search value in the Search field. Valid search fields are:

   - Name—Returns all applications with a name matching the value specified.
   - MSAS Instance Name—Returns all applications in an MSAS instance that matches the MSAS Instance Name specified.
   - Tags—Returns all applications that contain the tag matching the value specified.

   You can use percent `%` as a wildcard, any place in the name. Asterisk `*` is not recognized as a wildcard and is treated as plain text. Searches using the Name and Tag field are case insensitive, but searches using the MSAS Instance Name field are case sensitive.

3. To search for applications by type, or to further refine the results from the search menu, select the type of applications for which you want to search and display in the results from the **Type** menu. Valid options are:

   - **Virtual Applications**—Applications defined in the MSAS environment that specify virtual URLs for back-end URLs. In this case MSAS acts as a reverse proxy.
   - **Proxy Applications**—Applications defined in the MSAS environment that specify back-end URLs that are proxied directly through the Mobile Security Access Server. In this case MSAS acts as a forward proxy.
   - **Direct URL**—Reserved application that defines URLs that can be directly accessed and are not intercepted by the Mobile Security Access Server.

- **Blocked URL**—Reserved application that defines URLs that are designated as inaccessible, or blacklisted.
- **ALL**—All types of applications in the environment.

  For more information, see "Mobile Security Access Server Application Types" on page 3-1.

4. From the **Sort By** menu, select the order to display the search results. You can sort by Application name, MSAS Instance Name, or by applications that were modified most recently.

3.8 Viewing MSAS Application Details

To view the details for an MSAS application:

1. Navigate to the MSAS Applications page:
   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.
   c. If necessary, narrow the list of applications displayed by using the Search field. See "Searching for MSAS Applications" on page 3-7.
2. Click the icon, or the name of the application for which you want to view the details.
3. Use the MSAS Application Details page to:
   - View summary information about the application such as the associated instance, security context, and when it was last modified.
   - View the number of URLs configured in the application. Click the URLs search icon to navigate to the URLs or Proxy URLs page to view details about the URLs. See "Managing URLs in an MSAS Application" on page 3-9.
   - Navigate to the Application Roles page where you can view or search for configured roles, add or delete roles, and configure role hierarchy. See "Managing Roles in an MSAS Application" on page 5-9.
   - View and edit the tags configured in the application.You can use tags to categorize applications to make them easier to locate in the console. See "Categorizing MSAS Applications Using Tags" on page 3-11.
   - Export the application.

3.9 Editing an MSAS Application

After you create an MSAS application, you can add or edit URLs, application roles, and tags in the application.

To edit an MSAS application:

1. Navigate to the MSAS Applications page:
   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed by using the Search field. See "Searching for MSAS Applications" on page 3-7.

2. Click the icon, or the name of the application that you want to edit.
3. In the MSAS Application Details page, you can edit the application as follows:
   - Click the URLs search icon to navigate to the URLs or Proxy URLs page. From this page you can search for URLs in the application, view details about the URLs, edit the URL definition, add or delete URLs, and navigate to the URL Policy Configuration page. See "Managing URLs in an MSAS Application" on page 3-9.
   - Click the Application Roles search icon to navigate to the Application Roles page where you can view or search for configured roles, and add roles. See
   - Click the tags icon to add or edit tags in the application.You can use tags to categorize applications to make them easier to locate in the console. See "Categorizing MSAS Applications Using Tags" on page 3-11.
4. When you have finished editing the application, click **Apply** to save your changes.

> **Note:** When an application is being edited, the name in the tab is shown in italics. When you click Apply or Revert to save or discard the changes, the font returns to normal.

3.10 Managing URLs in an MSAS Application

You can manage the URLs in an application from the Proxy URLs or URLs page (virtual URLs). From this page you can search for URLs in the application, view details about the URLs, edit the URL definition, add or delete URLs, and navigate to the URL Policy Configuration page where you can attach policies and assertions to secure the URLs.

To manage the URLs in an MSAS application, navigate to the URLs or Proxy URLs page:

1. From the Mobile Security Launch Pad in the OAM console, click **Applications** in the Mobile Security Access Server section. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.
2. If necessary, narrow the list of applications displayed on the MSAS Applications page using the Search field. See "Searching for MSAS Applications" on page 3-7.
3. Click the application type icon, or the name of the application for which you want to manage the URLs to open the MSAS Application Detail page. The number of URLs configured in the application is displayed.
4. Click the URLs search icon to open the URLs or Proxy URLs page.
5. Use the URLs or Proxy URLs page to:
   - View a list of the virtual or proxy URLS configured in the application.
   - Search for virtual or proxy URLs in the application. Enter all or part of the URL name in the **Search** field and press the search icon.
   - Add virtual or proxy URLs to an application. Click **+Proxy URL** or **+URL** (for virtual applications) and complete the fields. For more information about adding URLs to an application, see "Creating a Virtual Application" on page 3-3 and "Creating a Proxy Application" on page 3-5.

- Delete virtual or proxy URLs from an application. Click the **Options** menu icon then **Delete** in the row containing the URL to be deleted.
- Edit the definition of a URL. Click the **Options** menu icon, then **Edit**, in the row containing the URL to be edited. In the pop-up window, edit the fields as required and click **Apply**.
- Navigate to a page where you can view or edit the security configuration of a URL using policies and assertions. Click the URL icon or the URL name to open the URL Policy Configuration page for the URL in a new tab. From this page, you can secure the URL at the policy enforcement points. For details, see Chapter 5, "Securing Mobile Security Access Server Resources."

3.11 Exporting MSAS Applications

You can export an MSAS application in a zip archive to be used in a different MSAS environment. Used in combination with Import, you can move applications between repositories. You export applications from the MSAS Applications page, or the MSAS Application Details page for a specific application.

> **Note:** You cannot export the `BLOCK`, `DIRECT`, `msm`, and `msm-reverse-proxy` reserved applications.
>
> Also, you should not export the `OAM Pass-through Proxy App` reserved application.

To export an MSAS application:

1. Navigate to the MSAS Applications page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.

2. In the MSAS Applications page, click the Options menu icon in the row for the application to be exported and click **Export**. Alternatively, click the icon or the name of the application in the table then, on the MSAS Application Details page, click **Export**.

3. Save the zip archive to your file system.

The directory structure for each application is maintained in the archive file using the following structure:

```
META-INF/virtualapplication/MSASInstanceName/application_name
```

3.12 Importing MSAS Applications

You can use the import feature to import a zip archive containing an MSAS application into your environment. Used in combination with **Export**, you can move applications between different repositories.

To import an MSAS application:

1. Navigate to the MSAS Applications page.

    a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

    b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

2. Click **Import**.

3. In the Import MSAS Application window: l

    a. Locate the zip archive to be imported on your local file system.

    b. Select the MSAS instance to which you want to import the application.

    c. Click **Import**. The imported application is added to the list of applications in the Applications table.

> **Notes:** All application names in an instance must be unique. If you attempt to import an application with the same name as an application that already exists in the instance, you are prompted to select an alternate zip file.
>
> The applications to be imported must use the following directory structure:
>
> `META-INF/virtualapplication/`*`MSASInstanceName/application_name`*

3.13 Categorizing MSAS Applications Using Tags

Tags provide a way to categorize applications to make them easier to find in the console. After you create an application, you can edit the application to add tags. You can then use the tags in the search field in the MSAS Applications page to quickly find applications with the assigned tag.

To use tags in an application:

1. Navigate to the MSAS Applications page:

    a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

    b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

    c. If necessary, narrow the list of applications displayed by using the Search field. See "Searching for MSAS Applications" on page 3-7.

2. Click the icon, or the name of the application to which you want to add tags.

3. In the MSAS Application Details page, click the Tags icon to open the Tags window.

4. To add, edit, or delete a tag:

    - To add a tag, click **Add** and enter the tag name in the Tag field.
    - To delete a tag, click the **X** in the row containing the tag to be deleted.
    - To edit a tag, click the tag name and edit the tag name in the Tag field as required.

5. Click **OK** to close the Tags window and click **Apply** to save the changes in the application.

3.14 Deleting MSAS Applications

You can delete MSAS applications from the MSAS Applications page.

> **Note:** You cannot delete the `BLOCK`, `DIRECT`, `msm`, and `msm-reverse-proxy` reserved applications.
>
> Also, you should not delete the `OAM Pass-through Proxy App` reserved application.

To delete an MSAS application:

1. Navigate to the MSAS Applications page:
   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.
   c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.
2. In the MSAS Applications page, click the Options menu icon in the row for the application to be deleted and click **Delete**.
3. In the Delete Application window, click **Delete** to confirm the deletion.

   The application is removed from the table in the MSAS Applications page.

4

Configuring Web Settings in MSAS

This chapter introduces the proxy auto-config (PAC) file in MSAS and describes how to configure blocked, direct, and proxy URL web settings in Mobile Security Access Server. The web settings that are defined in these PAC files instruct the Secure Workspace app and containerized apps which URLs to proxy through MSAS, which URLs to access directly without going through MSAS, and which URLs to block completely.

It includes the following sections:

4.1 Understanding PAC Files in MSAS

When dealing with a large set of URLs, the industry has defined the concept of a proxy auto-config (PAC) file. The PAC file allows clients (browsers, user agents, mobile devices, and so on) to be automatically configured in terms of how the URLs should be accessed. Mobile Security Access Server (MSAS) can act as a proxy or a reverse proxy as described in "Getting Started with Mobile Security Access Server" on page 1-1, and throughout this book. It generates a PAC file automatically based on the how the URLs in an MSAS instance are configured.

A PAC file includes a Java Script function `FindProxyForURL(`*`url,host`*`)` that determines how the URLs are to be proxied through MSAS, such as proxied, directly, or blocked.

Each MSAS instance generates two PAC file accessible at the following URLs:

- `https://`*`msas-host`*`:`*`msas-ssl-port`*`/bmax/stunnel.pac`
- `http://`*`msas-host`*`:`*`msas-port`*`/bmax/bmax.pac`

where *`msas-host`*`:`*`msas-ssl-port`* represents the host and SSL port of the MSAS instance, and *`msas-host`*`:`*`msas-port`* represents the host and the non-SSL port of the MSAS instance. The Secure Workspace app is automatically configured with these files during registration and then enforces the rules defined in them.

The only difference between these two PAC files is the port that they use for forward proxy calls. The `stunnel.pac` file uses the HTTPS port and is used by the Secure Workspace and containerized apps for forward proxy calls. The `bmax.pac` file uses the HTTP port and is used by browsers such as Safari for forward proxy calls.

The stunnel.pac and bmax.pac files are refreshed periodically to reflect any changes made to web settings. By default, the refresh interval is 24 hours. You can configure the interval using the cache.refresh.repeat property or force an immediate refresh using the synchronize feature as described in "Synchronizing MSAS Instance Configuration" on page 2-2. For details about configuring the cache.refresh.repeat property, see:

- "Configuring the Cache Refresh Time" on page 6-13
- "Configuring the Cache Refresh Time Using WLST" on page 6-49

4.2 Understanding the Mapping Between PAC Files and MSAS Applications

When you create an MSAS instance, DIRECT and BLOCK applications are created by default. You edit these applications to configure the direct and blocked URLs. These applications are reserved and cannot be deleted.

A Default URL proxy application is also created. This application is provided for convenience purposes and is used to secure all forward-proxy requests by default. It contains a wildcard path (/) that applies to all URLs in the environment that are not defined explicitly in a proxy application. By using a default URL, you do not need to define every URL in the system using a proxy URL. You can edit the URL defined in the application, and you can add or delete proxy URLs. You can edit the name of this application, and delete it.

You can create proxy applications in MSAS that specify back-end URLs that will be proxied directly through the Mobile Security Access Server. Any URL defined within a proxy application is automatically classified with the "proxy" access type. Proxy applications can be updated or deleted.

Table 4–1 describes the mapping between MSAS applications and the corresponding PAC files that are generated.

Table 4–1 Mapping Between MSAS Applications and the Proxy Auto-Config File

Application Type	PAC File Mapping
MSAS Proxy Application	All URLs within MSAS Proxy Apps are handled as Proxy URLs in PAC. This instructs the client that all matching URLs in this section should be proxied through the MSAS.
MSAS DIRECT Application	All URLs within this reserved application map to Direct URLs in PAC. This instructs the client that all matching URLs in this section should be accessed directly without going through MSAS.
MSAS BLOCK Application	All URLs within this reserved application map to Block URLs in PAC. This instructs the client that all matching URLs in this section should be completely blocked from access.

MSAS performs string matching on the URLs as follows:

- Each URL must include the scheme, and can optionally include the host, path, or query string, or any parts of those components. You can use the wildcard * for specific URL patterns as follows:
  - For HTTP and HTTPS requests, use http://* and https://*.
  - For any request hitting a specific domain, for example *.example.com.
  - For any request hitting a specific domain only over HTTPS, for example https://*.example.com.

   - To specify parts of path but not others. For example, configuring `https://www.example.com/somedir/*` in the BLOCK application will block access to all sub-paths under `somedir`. Configuring `https://www.example.com/somedir/stock/*` in a proxy application will allow access for all sub-paths under `somedir/stock`.

- All URLs are converted to lowercase before matching.
- By default, if a requested URL does not match any entries in the access list, and there is no default proxy URL defined, then the requested URL is sent as a direct type.

4.3 Configuring Blocked URLs

To configure blocked URLs, add them to the reserved BLOCK application for the MSAS instance. To do so:

1. From the Mobile Security **Launch Pad**, select **Applications** in the Mobile Security Access Server section.
2. Use the **Search** feature to locate the BLOCK application for the instance. Either of the following methods will work:
   - From the **Type** menu, select **Blocked URLs**. A list of the first five BLOCK applications in the environment, sorted as shown in the **Sort By** field, is displayed. Click **Load More Items** to display additional applications.
   - From the **Search** menu, select **MSAS Instance Name**, enter all or part of the instance name in the Search field and click the search icon. `%` is accepted as a wildcard in the search field. If the BLOCK application for the desired MSAS instance isn't displayed, select **Blocked URLs** from the **Type** menu.
3. Click the Block icon or the BLOCK application name to open the Blocked URLs page.
4. Click **+URL** to open the Add Blocked URL page. If no URLs are currently defined in the application, you can also click **Create URL**.
5. In the Add Blocked URL page, click **Add**.
6. Complete the **Host URL** and **Name** fields, and optionally, the **Description** field.
7. Continue to click **Add** and complete the fields for all URLs to be blocked.
8. Click **Save**.

4.4 Configuring Direct URLs

To configure direct URLs, add them to the reserved DIRECT application for the MSAS instance. To do so:

1. From the Mobile Security **Launch Pad**, select **Applications** in the Mobile Security Access Server section.
2. Use the **Search** feature to locate the DIRECT application for the instance. Either of the following methods will work:
   - From the **Type** menu, select **Direct URLs**. A list of the first five DIRECT applications in the environment, sorted as shown in the **Sort By** field, is displayed. Click **Load More Items** to display additional applications.
   - From the **Search** menu, select **MSAS Instance Name**, enter all or part of the instance name in the Search field and click the search icon. `%` is accepted as a

wildcard in the search field. If the DIRECT application for the desired MSAS instance isn't displayed, select **Direct URLs** from the **Type** menu.

3. Click the Direct icon or the Direct application name to open the Direct URLs page.
4. Click **+URL** to open the Add Direct URL page. If no URLs are currently defined in the application, you can also click **Create URL**.
5. In the Add Direct URL page, click **Add**.
6. Complete the **Host URL** and **Name** fields, and optionally, the **Description** field.
7. Continue to click **Add** and complete the fields for all URLs to be given direct access.
8. Click Save.

5 Securing Mobile Security Access Server Resources

Mobile Security Access Server (MSAS) provides a central access point for securing traffic from mobile devices to intranet resources such as web sites or web services exposed as URLs. These resources, defined as URLs inside MSAS proxy and virtual applications, can be secured using predefined access policies and assertions installed with MSAS.

This chapter includes the following topics:

5.1 Overview of Mobile Security Access Server Resource Security

To secure the communication between the mobile device and the back-end URLs, you can attach access policies and assertions at policy enforcement points on each URL in an MSAS application. These policies and assertions enforce security during the request phase from the client, the invocation of the back-end URL, and the response phase back to the client.

You can configure both authentication and authorization using the access policies. Typically, the authorization policies work in conjunction with the authentication policies. The authentication policies are used to verify the identity of the user accessing the URL, then the authorization policy confirms the roles to which the user belongs and performs the authorization check.

The predefined policies provided with Mobile Security Access Server are based on common best practice policy patterns used in customer deployments. The policies are constructed using assertions based on predefined assertion templates. If a predefined policy satisfies the requirements of your use case, you can attach the policy to the appropriate endpoints. If, however, you need to edit and customize the policy for each

URL, you may prefer to directly attach an assertion, provided in a predefined assertion template, because you can edit the attached assertion directly.

5.2 Attaching Policies and Assertions to Virtual Applications

Virtual applications defined in the MSAS environment specify virtual URLs for back-end URLs. In this case, the Mobile Security Access Server acts as reverse-proxy and hides the actual back-end URL from the clients.

In a virtual application, you attach policies and assertions at the HTTP method level of virtualized URLs. The policy enforcement endpoints to which you can attach policies are:

- On-Request—Secures the request phase from the client to Mobile Security Access Server using the policies and assertions referenced. Note that the same policies and assertions that are attached to the on-request endpoint are automatically attached to the on-response endpoint to secure the response sent back to the client.
- Invoke—Secures the connection between Mobile Security Access Server and the back-end service using the policies and assertions referenced.
- On-Response—Secures the response phase from Mobile Security Access Server to the client. If the attached policy or assertion does not have any response behavior, the policy is not enforced.

For a list of the policies and assertions that are supported for MSAS, see "Summary of Supported Policies and Assertions" on page 5-16.

To attach policies and assertions to the policy enforcement endpoints of a virtual URL:

1. Navigate to the URLs Summary page for the virtual application to be secured:
   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.
   c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.
   d. Click the virtual application icon, , or the name of the virtual application that you want to secure.
   e. In the MSAS Application Details page, click the URLs search icon to display the URLs Summary page for the virtual application.
2. Click the URL icon, or the name of the URL to be secured to display the URL Policy Configuration page.
3. Click the options menu for the policy enforcement endpoint to which you want to attach policies or assertions.
4. To attach a policy to a policy enforcement endpoint:
   a. Click **Add Policy** to attach a policy. The list of policies is filtered to reflect only the policies that are available for the type of endpoint. For example, only policies that can be attached to invoke the back-end service are listed for the Invoke policy enforcement endpoint.

      You can use the search field to refine the list of policies displayed. Enter all or part of a policy name in the Search field, select the desired operator, and press the search icon.The results that match the search criteria are displayed in the Search Results table.

Optionally, use the View menu to change the columns displayed, or to change the order of the columns.

b. Select the policy to be attached and click **Add Selected**.

c. Review the selections in the Selected Policies table. To remove an entry from the table, select the policy to be removed, then click **Remove Selected**. To remove all policies from the table, click **Remove All**.

   When you have confirmed the list of policies to be attached, click **Attach Policies**. The attached policies are listed in a table under the associated policy enforcement endpoint.

5. To attach an assertion to a policy enforcement endpoint:

   a. Click **Add Assertion** to attach an assertion to the endpoint. The list of assertions is filtered to reflect only the assertions that are available for the type of endpoint, organized by template name. For example, only assertions that can be attached to invoke the back-end service are listed for the Invoke policy enforcement endpoint.

      You can use the search field to refine the list of assertions displayed. Enter all or part of an assertion name in the Search field, select the desired operator, and press the search icon.The results that match the search criteria are displayed in the Search Results table.

      Optionally, use the View menu to change the columns displayed, or to change the order of the columns.

   b. Select the assertion to be attached and click **Add Selected**.

   c. Review the selections in the Selected Assertion Templates table. To remove an assertion from the table, select the assertion to be removed, then click **Remove Selected**. To remove all assertions from the table, click **Remove All**.

      If desired, you can also specify an alternate name for the assertion in the **Assertion Name** field.

   d. When you have confirmed the assertions to be attached, click **Add Assertion**. The attached assertions are listed in a table under the associated policy enforcement endpoint.

6. To change the order of the policies and assertions attached to a policy enforcement endpoint, click the options menu ≡ for the endpoint, then click **Reorder**. In the Reorder window, select the policy or assertion, then click the up or down arrow to adjust the order as desired. Click **OK**.

   Policies and assertions are enforced in the order in which they are attached to the endpoint.

7. Click **Validate** to ensure that the combination of policies and assertions attached to the endpoint is valid.

   If there is a validation error, a dialog box displays describing the error. To remove an attached policy/assertion template, click the options menu for the policy or template, then click **Delete**.

8. Click **Apply** to attach the policies or assertions to the endpoint.

   Note that the same policies and assertions that are attached to the on-request endpoint are automatically attached to the on-response endpoint to secure the response sent back to the client. If the attached policy/assertion does not support response behavior, then the attachment is ignored.

5.3 Attaching Policies and Assertions to Proxy Applications

Proxy applications defined in the MSAS environment specify back-end URLs that will be proxied directly through the Mobile Security Access Server. In this case, the Mobile Security Access Server acts as a forward proxy. The back-end URLs are visible to the client but the requests are proxied through the Mobile Security Access Server.

For proxy applications, you attach policies at the URL level corresponding to the back-end URL. The policy enforcement endpoints to which you can attach policies are:

- On-Request—Secures the request phase from the client to Mobile Security Access Server using the policies and assertions referenced. Note that the same policies and assertions that are attached to the on-request endpoint are automatically attached to the on-response endpoint to secure the response sent back to the client.
- Invoke-Proxy—Secures the connection between Mobile Security Access Server and the back-end service using the policies and assertions referenced.
- On-Response—Secures the response phase from Mobile Security Access Server to the client. If the attached policy or assertion does not have any response behavior, the policy is not enforced.

For a list of the policies and assertions that are supported for MSAS, see "Summary of Supported Policies and Assertions" on page 5-16.

To attach policies and assertions to the policy enforcement endpoints of a proxy URL:

1. Navigate to the URLs Summary page for the proxy application to be secured:
   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.
   c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.
   d. Click the proxy application icon, , or the name of the proxy application that you want to secure.
   e. In the MSAS Application Details page, click the URLs search icon to display the Proxy URLs Summary page for the proxy application.
2. Click the Proxy URL icon, , or the name of the URL to be secured to display the URL Policy Configuration page.
3. Click the options menu ☰ for the policy enforcement endpoint to which you want to attach policies or assertions.
4. To attach a policy to a policy enforcement endpoint:
   a. Click **Add Policy** to attach a policy. The list of policies is filtered to reflect only the policies that are available for the type of endpoint. For example, only policies that can be attached to invoke the back-end service are listed for the Invoke-Proxy policy enforcement endpoint.

      You can use the search field to refine the list of policies displayed. Enter all or part of a policy name in the Search field, select the desired operator, and press the search icon.The results that match the search criteria are displayed in the Search Results table.

   Optionally, use the View menu to change the columns displayed, or to change the order of the columns.

   b. Select the policy to be attached and click **Add Selected**.

   c. Review the selections in the Selected Policies table. To remove an entry from the table, select the policy to be removed, then click **Remove Selected**. To remove all policies from the table, click **Remove All**. Note that if you are adding an assertion, you can also specify an alternate name for the assertion in the Assertion Name field.

   When you have confirmed the list of policies to be attached, click **Attach Policies**.

5. To attach an assertion to a policy enforcement endpoint:

   a. Click **Add Assertion** to attach an assertion to the endpoint. The list of assertions is filtered to reflect only the assertions that are available for the type of endpoint, organized by template name. For example, only assertions that can be attached to invoke the back-end service are listed for the Invoke policy enforcement endpoint.

   You can use the search field to refine the list of assertions displayed. Enter all or part of an assertion name in the Search field, select the desired operator, and press the search icon.The results that match the search criteria are displayed in the Search Results table.

   Optionally, use the View menu to change the columns displayed, or to change the order of the columns.

   b. Select the assertion to be attached and click **Add Selected**.

   c. Review the selections in the Selected Assertion Templates table. To remove an assertion from the table, select the assertion to be removed, then click **Remove Selected**. To remove all assertions from the table, click **Remove All**.

   If desired, you can also specify an alternate name for the assertion in the **Assertion Name** field.

   d. When you have confirmed the assertions to be attached, click **Add Assertion**.

6. To change the order of the policies and assertions attached to a policy enforcement endpoint, click the options menu ≡ for the endpoint, then click **Reorder**. In the Reorder window, select the policy or assertion, then click the up or down arrow to adjust the order as desired. Click **OK**.

   Policies and assertions are enforced in the order in which they are attached to the endpoint.

7. Click **Validate** to ensure that the combination of policies and assertions attached to the endpoint is valid.

   If there is a validation error, a dialog box displays describing the error. To remove an attached policy/assertion template, click the options menu for the policy or template, then click **Delete**.

8. Click **Apply** to save the attachments to the URL.

   Note that the same policies and assertions that are attached to the on-request endpoint are automatically attached to the on-response endpoint to secure the response sent back to the client. If the attached policy/assertion does not support response behavior, then the attachment is ignored.

5.4 Viewing Policies Attached to an Application

You can view the policies attached to each URL in an application on the URL Policy Configuration page for the URL.

To view the policies attached to a URL in an application:

1. Navigate to the MSAS Applications page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed by using the Search field. See "Searching for MSAS Applications" on page 3-7.

2. Click the icon, or the name of the application for which you want to view the attached policies.

3. Click the URLs search icon to navigate to the URLs page for virtual applications or Proxy URLs page for proxy applications.

4. Click the URL or Proxy URL icon, or the URL name, to open the URL Policy Configuration page. The attached policies and assertions are listed in a table under the associated policy enforcement endpoint.

5. To view the details of an attached policy, select the policy in the endpoint table. Policies are indicated with the 📋 icon.

   The policy details are displayed in the right pane. Details are provided in the following tabs.

Tab	Description
General	For attached policies, this tab displays general information about the policy in read-only format, including the name, display name, category, description, whether the policy is enabled, and the type of endpoints to which the policy can be attached, and version information for the policy. Click **Versioning History** to open the Policy Version History page that you use to view a list of all versions of the policy, view the details of any policy version in read-only format, activate any version of a policy, and delete or export any version of a policy. You cannot edit a policy from the Policy Version History page. You must edit and save the policy in the Policy Details page. For more information, see "Creating and Editing a Policy" on page 8-5.
Assertions	Click this tab to view the assertions in the policy. Click an assertion to view details about the assertion including the name, the category to which the assertion belongs (for example security/authentication or security/authorization), the type of assertion (for example http-jwt-token), and whether the assertion is enforced. The Details section provides the ability to view the settings for the selected assertion. Assertion template details vary based on the type of assertion. For example, assertions that include message protection will include settings that are specific to message security.

Tab	Description
Overrides	Click this tab to view the configuration properties for the policy. Configuration properties vary based on the assertion in the policy. Use these fields to override a property on a per-attachment basis. For details about overriding configuration properties in a policy, see "Configuring Policy Overrides" on page 5-7.
Authorization	**Note:** This tab is available only with the Fine-grained authorization using Oracle Entitlements Server policy and is used to configure fine-grained authorization. For configuration information, see "Configuring Authorization" on page 5-14.

6. To view the details of an attached assertion, select the assertion in the endpoint table. Assertions are indicated with the icon.

   The assertion details are displayed in the right pane. Details are provided in the following tabs.

Tab	Description
General	This tab displays details about the assertion including the name, the category to which the assertion belongs (for example security/authentication or security/authorization), the type of assertion (for example http-jwt-token), and whether the assertion is enforced. The Details or Settings section provides the ability to view the settings for the selected assertion. Assertion template details vary based on the type of assertion. For example, assertions that include message protection will include settings that are specific to message security. Details about each predefined policy and assertion template are provided in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.
Overrides	Click this tab to view the configuration properties for the assertion. Configuration properties vary based on the assertion. Use these fields to override a property on a per-attachment basis. For details about overriding configuration properties in a policy, see "Configuring Policy Overrides" on page 5-7.

5.5 Configuring Policy Overrides

The configuration properties included with the predefined policies allow you to override certain configuration settings, such as the CSF key used for storing the signature-key password. The scope for the configuration property override value is limited to the specific policy attachment.

You can override the value when you attach the policy to the endpoint as described in this section. Alternatively, you can change the default value of a configuration override property in a policy. If you do so, any endpoint to which you attach the policy can use these values. To edit the configuration property in a policy, see "Creating and Editing a Policy" on page 8-5.

> **Note:** Oracle recommends that you do not edit the predefined policies so that you will always have a known set of valid policies.
>
> If you wish to edit a configuration property in a predefined policy, Oracle recommends that you clone the policy and then edit it.

The configuration properties that you can override in a predefined policy are inherited from the assertion templates that are included in the policy. For additional information, see the following topics:

- "Predefined Policies" and "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server* to determine the configuration properties associated with each policy and assertion template.
- "Assertion Template Configuration Properties" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server* for an alphabetized list of the overrideable properties.
- "Predefined Policies" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for additional policies and associated configuration properties.

To override configuration properties in an attached policy or assertion:

1. Attach the policy or assertion template to the endpoint as described in "Attaching Policies and Assertions to Virtual Applications" on page 5-2 and "Attaching Policies and Assertions to Proxy Applications" on page 5-4.
2. Select the policy or template from the list under the policy enforcement endpoint. The details for the policy or template are displayed in the right pane.
3. Click the **Overrides** tab.
4. Enter the override value in the **Value** field for the property and click **Apply**.

   The property is overridden on a per-attachment basis.

Note: You cannot override a property of type "constant".

5.6 Validating Policy Attachments

The type and number of assertions within a policy may be valid and, therefore, a policy may be internally consistent and valid. However, when more than one policy is attached to a policy enforcement endpoint, the combination of policies must also be valid. Specifically, the following must be true:

- If the On-Request policy enforcement endpoint contains an authentication policy and an authorization policy, the authentication policy must precede the authorization policy. Instructions for reordering policy attachments are provided in "Attaching Policies and Assertions to Virtual Applications" on page 5-2 and "Attaching Policies and Assertions to Proxy Applications" on page 5-4.
- Only one security policy or assertion with subtype authentication can be attached to a policy enforcement endpoint.

5.7 Detaching Policies from an Application

To detach a policy from a virtual or proxy application:

1. Navigate to the MSAS Applications page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed by using the Search field. See "Searching for MSAS Applications" on page 3-7.

2. Click the icon, or the name of the application for which you want to detach a policy or assertion.
3. Click the URLs search icon to navigate to the URLs page for virtual applications or Proxy URLs page for proxy applications.
4. Click the URL or Proxy URL icon, or the URL name, to open the URL Policy Configuration page. The attached policies and assertions are listed in a table under the associated policy enforcement endpoint.
5. To detach a policy or assertion, click the options icon for the policy or assertion, then click **Delete**.
6. Click **OK** in the Delete window.
7. Click **Apply** to save your changes.

5.8 Configuring Authorization in MSAS Applications

Authorization (also known as access control) is granting access to specific resources based on an authenticated user's entitlements. Entitlements are defined by one or several attributes. An attribute is the property or characteristic of a user, for example, if "Marc" is the user, "conference speaker" is the attribute.

Most often, authentication is the first step of determining whether a user should be given access to a resource. After the user is authenticated, the second step is to verify that the user is authorized to access the resource.

Authorization enables you to determine what operations authenticated clients can access. Mobile Security Access Server uses role-based authorization, which is based on the notion that a set of identities, known as principals, can be grouped into roles, and then a policy can be applied to each of the roles.

The following sections describe how to configure authorization in Mobile Security Access Server:

- Managing Roles in an MSAS Application
- Configuring Authorization

5.8.1 Managing Roles in an MSAS Application

In Mobile Security Access Server, the scope of an application role is the MSAS application. That is, the roles in one MSAS application apply only to that application and are not visible to other MSAS applications. Application roles are supported in both virtual and proxy applications.

Mobile Security Access Server provides the following functions for application role management:

- Create, update, delete, and view application roles.
- Manage application role hierarchy where roles can be combined in a hierarchy where higher-level roles subsume permissions owned by sub-roles.
- Map external roles to application roles.
- Map users to application roles.

You manage roles in a MSAS application using the MSAS Application Roles Summary page. From this page you can:

- View a list of the application roles configured in the application.
- Search for application roles in the application.
- Navigate to the Application Roles page where you can create and add roles to an application, edit existing roles, manage application role hierarchy, and map users to application roles.
- Delete application roles from an application.

5.8.1.1 Creating an Application Role

You create MSAS application roles from the Application Roles Summary page.

To create an MSAS application role:

1. Navigate to the Applications Roles Summary page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.

   d. Click the application icon, or the name of the application for which you want to create roles.

   e. In the MSAS Application Details page, click the Application Roles search icon.

2. In the Application Roles Summary page, click **+Add Role**.

3. In the Application Roles page, enter a name, display name, and description for the application role.

Field	Description
Name	Enter a name for the application role. The name must be unique within the MSAS application.
Display Name	Optionally, enter a meaningful name that can be used to identify the application role in the console. If you do not provide a display name, the role name is used.
Description	Optionally, enter a brief description of the role.

4. Optionally, position the role being created into the application role hierarchy.

   a. Click the **App Role Hierarchy** tab if it is not already selected.

   b. Click **Inherits From** to specify the application roles from which the role being created should inherit permissions.

   c. Enter all or part of a role name in the Search field and click **Search**. An empty strings fetches all roles in the application. The first five roles are shown.

   d. Click **Load More Items** to display additional roles.

   e. Click **Add to Hierarchy** for each role from which the current role should inherit permissions. The selected roles are added to the App Roles table.

   f. Click **Inherited By** to view the application roles that inherit the permissions of this role.

5. Optionally, map external roles to the application role being created.

   a. Click the **External Role Mapping** tab.

   b. Enter all or part of an external role name in the Search field and click **Search**. An empty strings fetches all roles in the application. If necessary, click **Load More Items** to display additional roles.

   c. Select the external role and click **Map to Role** for each external role that you want to map to the role being created.

6. Optionally, map users to the application role being created.

   a. Click the **User Mapping** tab.

   b. Enter all or part of a user name in the Search field and click **Search**. An empty strings fetches all users in the application. If necessary, click **Load More Items** to display additional users.

   c. Select the user and click **Map to Role** for each user that you want to map to the role being created.

7. Click **Apply** to create the role with specified role hierarchy and mapping.

5.8.1.2 Viewing Roles in an MSAS Application

You can view the roles configured in an MSAS application from the Application Roles Summary page. From there you can click on a specific role to view the configuration details of that role.

To view the configured application roles:

1. Navigate to the MSAS Applications page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed by using the Search field. See "Searching for MSAS Applications" on page 3-7.

2. Click the application icon, or the name of the application for which you want to view the details.

3. In the MSAS Application Details page, click the Application Roles search icon to navigate to the Application Roles Summary page where the roles configured in the application are displayed in a table. Only the first 5 roles configured are displayed. To view additional roles, click **Load More Items.**

4. Optionally, use the Search field to find specific roles by entering all or part of a role name in the **Application Roles** search field and clicking the **Search** icon. Wildcards are not recognized and are treated as plain text. Searches are case-insensitive.

5. Click the role icon, the role name, or the options menu then **Edit Role** to open the Application Roles page

6. Click the **App Role Hierarchy**, **External Role Mapping**, and **User Mapping** tabs to view the role hierarchy and mapping.

5.8.1.3 Managing the Application Role Hierarchy

You can view and modify the role hierarchy of an application role, specifically the hierarchy of application roles below and above a given application role. Roles below a given role inherit the permissions of the selected role; a given role inherits the permissions of roles above it.

5.8.1.3.1 Roles that an Application Role Inherits To view or modify the application role hierarchy below a given application role:

1. If it is already open, click the **App Roles** tab for the application to display the Application Roles Summary page.

   Otherwise, navigate to the Applications Roles Summary page as described in "Viewing Roles in an MSAS Application" on page 5-11.

2. Click the role icon, the role name, or the options menu then **Edit Role** to open the Application Roles page.
3. Click the **App Role Hierarchy** tab then **Inherits From**.

   The App Roles table on the right displays the application roles from which the selected role inherits permissions.

4. To add a role to the hierarchy below the current role:

   a. Enter all or part of a role name in the Search field and click **Search**. An empty strings fetches all roles in the application.

      The first five roles are shown. If necessary, click **Load More Items** to display additional roles.

   b. Click **Add to Hierarchy** for each role from which the current role should inherit permissions. The selected roles are added to the App Roles table.

5. To remove a role from the hierarchy, select the role in the App Roles list and click **Remove**.
6. Click **Apply** to update the role hierarchy.

5.8.1.3.2 Roles that Inherit an Application Role To view or modify the application role hierarchy above a given application role:

1. If it is already open, click the **App Roles** tab for the application to display the Application Roles Summary page.

   Otherwise, navigate to the Applications Roles Summary page as described in "Viewing Roles in an MSAS Application" on page 5-11.

2. Click the role icon, the role name, or the options menu then **Edit Role** to open the Application Roles page.
3. Click the **App Role Hierarchy** tab then **Inherited By**.

   The App Roles table displays the application roles that inherit the permissions of the current role.

5.8.1.4 Mapping External Roles to an Application Role

To map external roles to an application role:

1. If it is already open, click the **App Roles** tab for the application to display the Application Roles Summary page.

Otherwise, navigate to the Applications Roles Summary page as described in "Viewing Roles in an MSAS Application" on page 5-11.

2. Click the role icon, the role name, or the options menu then **Edit Role** to open the Application Roles page.
3. Click the **External Role Mapping** tab.
4. To map an external role to the current role:
   a. Enter all or part of a role name in the Search field and click **Search**. An empty strings fetches all roles in the application. If necessary, click **Load More Items** to display additional roles.
   b. Click **Map to Role** for each external role that you want to map to the current role. The selected roles are added to the Mapped Roles table.
5. To remove the mapping from an external role, select the role in the Mapped Roles list and click **Remove**.
6. Click **Apply** to update the role mapping.

5.8.1.5 Mapping Users to an Application Role

To map users to an application role:

1. If it is already open, click the **App Roles** tab for the application to display the Application Roles Summary page.

   Otherwise, navigate to the Applications Roles Summary page as described in "Viewing Roles in an MSAS Application" on page 5-11.
2. Click the role icon, the role name, or the options menu then **Edit Role** to open the Application Roles page.
3. Click the **User Mapping** tab.
4. To map users to the current role:
   a. Enter all or part of a user name in the Search field and click **Search**. An empty strings fetches all users in the application. If necessary, click **Load More Items** to display additional users.
   b. Click **Map to Role** for each user that you want to map to the current role. The selected roles are added to the Mapped Users list.
5. To remove the mapping from a user, select the user in the Mapped Users list and click **Remove**.
6. Click **Apply** to update the role mapping.

5.8.1.6 Deleting Application Roles

You can delete MSAS application roles from the Application Roles Summary page.

To delete an application role:

1. If it is already open, click the **App Roles** tab for the application to display the Application Roles Summary page.

   Otherwise, navigate to the Applications Roles Summary page as follows:
   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.

d. Click the application icon, or the name of the application for which you want to create roles.

e. In the MSAS Application Details page, click the Application Roles search icon.

2. In the Application Roles Summary page, click the options menu icon in the row for the role to be deleted and click **Delete Role**.

3. In the Remove App Role window, click **Remove** to confirm the deletion.

   The role is removed from the table in the Application Roles Summary page.

> **Note:** If the role that is deleted is inherited by another role, the inheritance is also deleted.

5.8.2 Configuring Authorization

You can configure authorization in Mobile Security Access Server by attaching the predefined authorization policy `oracle/binding_oes_authorization_policy` to the on-request policy enforcement point of a proxy or virtual URL, and then configuring the policy to specify users and roles that are authorized to access the proxy or virtual URL.

To configure authorization for an application:

1. Navigate to the URL Policy Configuration page for the URL for which you want to configure authorization:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section click **Applications**. Alternatively, you can click **Environments**, then click **Applications** in the MSAS tile.

   c. If necessary, narrow the list of applications displayed in the MSAS Applications page by using the Search field. See "Searching for MSAS Applications" on page 3-7.

   d. Click the application icon, or the name of the application.

   e. In the MSAS Application Details page, click the URLs search icon to display the URLs/Proxy URLs Summary page for the application.

   f. Click the URL/Proxy URL icon, or the name of the URL to be configured to display the URL Policy Configuration page.

2. Click the options menu ☰ for the On-Request policy enforcement endpoint and select **Add Policy**.

3. Select the `oracle/binding_oes_authorization_policy` from the table and click **Add Selected**, then **Attach Policies**.

> **Note:** Do not attach this policy to URLs that contain a wildcard `*`. If you do so the policy will not be enforced at run time.

4. Select the Fine-grained authorization using Oracle Entitlements Server policy in the table.

5. In the policy details in the right pane, select the **Authorization** tab.

6. In the Policy Effect field, select how the policy will govern the access to the protected URL:

   - **Permit**—Subjects will be permitted to access the protected URL.
   - **Deny**—Reserved for future use.

7. In the Subject field, specify the users for whom the operation selected in the Policy Effect field will be enforced. The subject can be configured as follows:

   - Application Role—The policy will be enforced for all members of the selected application roles.
   - External Role—The policy will be enforced for all members of the selected external roles.
   - User—The policy will be enforced for all selected users.

   To select the subjects:

   a. Click **+Add** to display the Add Roles page. By default, the Application Roles search operator is selected and the application roles configured in the application are listed in the Subject table. For details about configuring application roles, see "Managing Roles in an MSAS Application" on page 5-9.

   b. To add application roles to the subject list, select the roles in the table and click **Add Selected**.

   c. To add external roles to the subject list, enter all or part of a role name in the search **Name** field, select **External Role** from the menu, and click **Search**. An empty strings fetches all external roles in the application. If necessary, click **Load More Items** to display additional roles. Select the external roles to be added from the table and click **Add Selected**.

   d. To add users to the subject list, enter all or part of a user name in the search **Name** field, select **User** from the menu, and click **Search**. An empty strings fetches all users in the application. If necessary, click **Load More Items** to display additional users. Select the users to be added from the table and click **Add Selected**.]

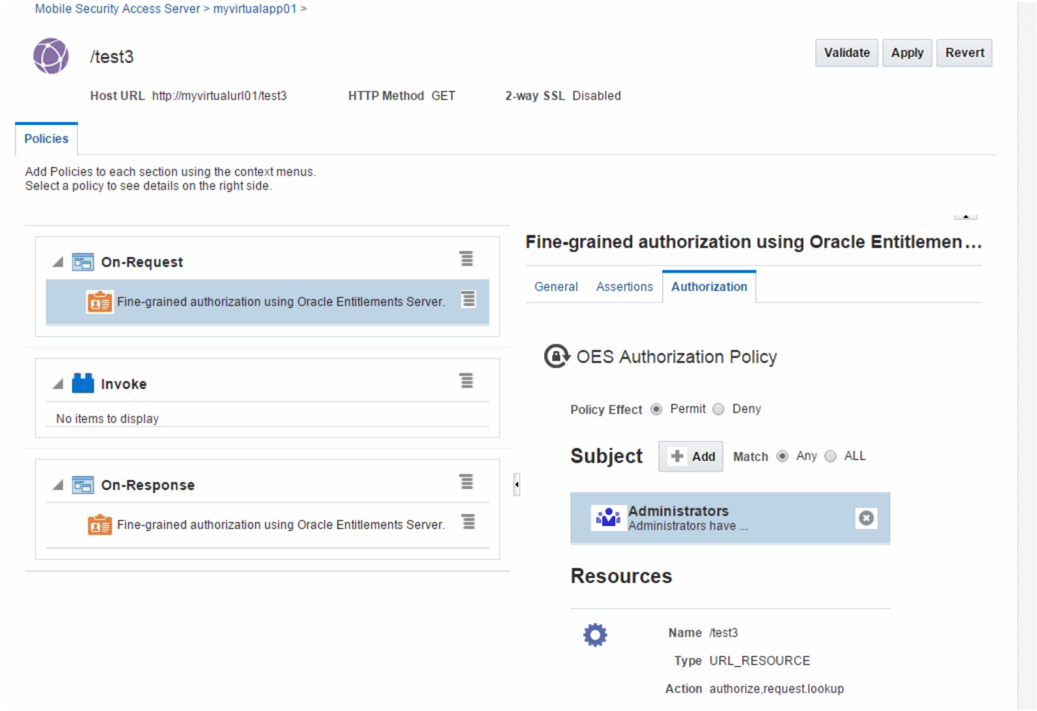
   e. Click **Add** to add the selected roles and users to the subject list.

8. If multiple subjects are specified, choose how the policy is enforced:

   - Any—The policy is enforced if at least one of the subject rules is satisfied.
   - All—The policy is enforced if all of the subject rules are satisfied.

   Figure 5–1 shows a virtual URL with an attached authorization policy configured to allow access to all members of the Administrators role.

Figure 5–1 OES Authorization Policy Configuration



/test3

Host URL http://myvirtualurl01/test3 HTTP Method GET 2-way SSL Disabled

Add Policies to each section using the context menus.
Select a policy to see details on the right side.

On-Request: Fine-grained authorization using Oracle Entitlements Server.

Invoke: No items to display

On-Response: Fine-grained authorization using Oracle Entitlements Server.

Fine-grained authorization using Oracle Entitlemen…

OES Authorization Policy

Policy Effect ● Permit ○ Deny

Subject Match ● Any ○ ALL

Administrators
Administrators have ...

Name /test3
Type URL_RESOURCE
Action authorize,request,lookup

9. Click **Apply** to save the authorization policy attachment and configuration. The authorization policy is automatically attached to the on-response endpoint. Because this policy does not support response behavior, the attachment is ignored.

> **Note:** You cannot edit a proxy URL that has the `oracle/binding_oes_authorization_policy` attached. Instead you should delete the URL and add a new proxy URL with the desired changes.

5.9 Summary of Supported Policies and Assertions

The following table summarizes the policies and assertions that are supported by Mobile Security Access Server based on policy enforcement endpoint type.

For more information about the policies and assertions, see:

- "Predefined Policies" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*
- "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*

Table 5–1 Supported Policies and Assertions

Policy Enforcement Endpoint	Supported Policies	Supported Assertions
On-request/On-response	■ oracle/binding_oes_authorization_policy ■ oracle/http_basic_auth_over_ssl_service_policy ■ oracle/http_jwt_token_over_ssl_service_policy ■ oracle/http_jwt_token_service_policy ■ oracle/http_oam_authentication_service_policy ■ oracle/http_saml20_token_bearer_over_ssl_service_policy ■ oracle/http_saml20_token_bearer_service_policy ■ oracle/http_session_token_verify_policy ■ oracle/wss_http_token_service_policy	■ oracle/binding_oes_authorization_template ■ oracle/http_jwt_token_over_ssl_service_template ■ oracle/http_jwt_token_service_template ■ oracle/http_oam_authentication_service_template ■ oracle/http_session_token_verify_template ■ oracle/wss_http_token_service_template
Invoke/Invoke-proxy	■ oracle/http_basic_auth_over_ssl_client_policy ■ oracle/http_bmax_jwt_user_token_client_policy ■ oracle/http_bmax_oam_client_policy ■ oracle/http_bmax_oauth_client_policy ■ oracle/http_bmax_spnego_client_policy ■ oracle/http_jwt_token_client_policy ■ oracle/http_jwt_token_over_ssl_client_policy ■ oracle/http_ntlm_token_client_policy ■ oracle/http_saml20_token_bearer_client_policy ■ oracle/http_saml20_token_bearer_over_ssl_client_policy ■ oracle/inject_header_with_client_certificate_policy ■ oracle/multi_token_client_policy ■ oracle/wss_http_token_client_policy	■ oracle/http_bmax_jwt_user_token_client_template ■ oracle/http_bmax_oam_client_template ■ oracle/http_bmax_oauth_client_template ■ oracle/http_bmax_spnego_client_template ■ oracle/http_jwt_token_client_template ■ oracle/http_jwt_token_over_ssl_client_template ■ oracle/http_ntlm_token_client_template ■ oracle/inject_header_template ■ oracle/wss_http_token_client_template

Table 5–1 (Cont.) Supported Policies and Assertions

Policy Enforcement Endpoint	Supported Policies	Supported Assertions
Internal Use Only	The following policies are used internally by MSAS to secure its instances. **Warning**: If you attach any of these policies to your policy enforcement endpoint, *your endpoint will not be secure*. Although the policy attachment will validate at design time, you will receive errors at run time. ▪ `oracle/http_action_over_ssl_policy` ▪ `oracle/http_form_based_auth_over_ssl_service_policy` ▪ `oracle/http_kinit_over_ssl_policy` ▪ `oracle/http_oauth2_confidential_client_over_ssl_policy` ▪ `oracle/http_oauth2_mobile_client_over_ssl_policy` ▪ `oracle/http_session_token_issue_policy` ▪ `oracle/http_pkinit_over_ssl_policy` ▪ `oracle/http_tlp_over_ssl_policy`	The following assertions are used internally by MSAS to secure its instances. **Warning**: If you attach any of these assertions to your policy enforcement endpoint, *your endpoint will not be secure*. Although the policy attachment will validate at design time, you will receive errors at run time. ▪ `oracle/http_form_based_auth_over_ssl_service_template` ▪ `oracle/http_session_token_issue_template`

The following table summarizes the policies and assertions that **are not** supported by MSAS based on policy enforcement endpoint type.

WARNING: If you attach any of the policies and assertions defined in Table 5–2 to your policy enforcement endpoint, *your endpoint will not be secure*. Although the policy attachment will validate at design time, you will receive errors at run time.

Table 5–2 Unsupported Policies and Assertions

Policy Enforcement Endpoint	Unsupported Policies	Unsupported Assertions
On-request/On-response	The following policies are reserved for future use: ■ `oracle/http_oam_token_service_policy` ■ `oracle/log_policy` ■ `oracle/multi_token_over_ssl_rest_service_policy` ■ `oracle/multi_token_rest_service_policy`	The following assertions are reserved for future use: ■ `oracle/http_oam_token_service_template` ■ `oracle/http_saml20_token_bearer_service_template` ■ `oracle/http_spnego_token_service_template` ■ `oracle/log_template` ■ `oracle/security_log_template` ■ `oracle/wss_http_token_over_ssl_service_template` ■ `oracle/xpath_token_auth_service_template` ■ `oracle/xpath_username_auth_service_template`
Invoke/Invoke-proxy	The following policies are reserved for future use: ■ `oracle/http_jwt_token_identity_switch_client_policy` ■ `oracle/http_oauth2_token_client_policy` ■ `oracle/http_oauth2_token_identity_switch_over_ssl_client_policy` ■ `oracle/http_oauth2_token_opc_oauth2_client_policy` ■ `oracle/http_oauth2_token_over_ssl_client_policy` ■ `oracle/log_policy` ■ `oracle/oauth2_config_client_policy` ■ `oracle/multi_token_over_ssl_client_policy`	The following assertions are reserved for future use: ■ `oracle/http_oauth2_token_client_template` ■ `oracle/http_oauth2_token_over_ssl_client_template` ■ `oracle/http_saml20_token_bearer_client_template` ■ `oracle/http_spnego_token_client_template` ■ `oracle/log_template` ■ `oracle/oauth2_config_client_template` ■ `oracle/security_log_template` ■ `oracle/wss_http_token_over_ssl_client_template`

6 Configuring a Mobile Security Access Server Instance

This chapter describes how to configure identity store profiles, trusted issuers and DNs, message security, policy access, and authentication endpoints and other security settings for a Mobile Security Access Server (MSAS) instance. You can perform this configuration using either the MSAS console component of the OAM console or WLST. Both methods are described. It also provides advanced configuration procedures for Kerberos, MSAS as a WebGate, and MSAS SSO, and OAuth2 client configuration.

Topics include:

- Understanding MSAS Instance Configuration
- Configuring an MSAS Instance Using the MSAS Console
- Configuring an MSAS Instance Using WLST
- Advanced Kerberos Configuration
- Manually Configuring OAuth2 Client Authentication
- Configuring Single Sign-On (SSO) for OAM WebGate and Oracle WSM Protected Resources
- Configuring Single Sign-On for Kerberos and NTLM Protected Resources
- Configuring an MSAS Instance as a WebGate

6.1 Understanding MSAS Instance Configuration

The Mobile Security Manager provides the capability for an administrator to configure and manage MSAS instances from a central location using the MSAS Instance Configuration page in the MSAS console component of the OAM console. From this page you can create identity store profiles, configure trusted issuers and message security settings, cache refresh rates, and authentication endpoints such as KINIT/PKINIT and OAuth2 confidential and mobile clients. You can also configure various system settings such as outbound message settings, proxy server settings, and logging, and others.

WLST commands also provide the ability to configure an MSAS instance from the command line or using scripts. For details about using the WLST commands, see "Configuring an MSAS Instance Using WLST" on page 6-27.

When you create an MSAS instance using the `configMSAS` script or the MSAS console, a default configuration document is created in the repository that contains all of the properties valid for the instance with their default values. When you modify the default configuration as described in this chapter, this configuration document is updated with the changes. For details about creating an MSAS instance using the `configMSAS` script, see "Configuring an MSAS Instance" in *Installing Oracle Mobile Security Access Server*.

The configuration performed using the MSAS console pages or the WLST commands applies at the logical instance level only. Any configuration defined for the instance is shared by all physical instances bound to the logical instance.

In most cases, changes made to the Mobile Security Access Server configuration do not require a server restart. By default, the changes are synchronized with the server using a user-defined polling interval, that you can specify using the cache refresh property. For details about setting this cache refresh property, see "Configuring the Cache Refresh Time" on page 6-13 and "Configuring the Cache Refresh Time Using WLST" on page 6-49. If you want these changes to go into effect immediately, you can use the synchronize feature as described in "Synchronizing MSAS Instance Configuration" on page 2-2.

In certain situations, however, such as creating an identity store profile, changes to the configuration do require a server restart. Table 6–1 lists the types of configuration updates that require you to restart the MSAS server.

Table 6–1 Configuration Changes that Require an MSAS Server Restart

Configuration Update	Additional information
Creating or editing an identity store profile	"Configuring the Identity Store Profile" on page 6-3 "Configuring an Identity Store Profile Using WLST" on page 6-31
Changes to SSL keystore and truststore	"Configuring the SSL Keystore and Truststore" on page 6-25 "Configuring SSL Settings Using WLST" on page 6-53
Setting outbound connection pool and HTTP(s) connection properties	"Configure Outbound Message Settings" on page 6-23 "Configuring Outbound Message Settings Using WLST" on page 6-52
Setting outbound proxy server settings	"Configure Proxy Server Settings" on page 6-24 "Configuring Proxy Server Settings Using WLST" on page 6-53

6.2 Configuring an MSAS Instance Using the MSAS Console

You manage the configuration of an MSAS instance using the MSAS Instance Configuration page. From this page you can view general configuration information about the instance, configure identity store profiles, define trusted issuers, configure keystore aliases and message settings, configure cache management, configure authentication endpoints, and other system settings.

The following sections describe how to configure an MSAS instance using the different tabs on the MSAS Instance Configuration page:

- Viewing General MSAS Instance Configuration
- Configuring the Identity Store Profile
- Configuring Trusted Issuers and DN Lists for Signing Certificates
- Configuring Message Security
- Configuring the Cache Refresh Time
- Configuring Authentication Endpoints
- Configuring System Settings

6.2.1 Viewing General MSAS Instance Configuration

The **General** tab of the MSAS Instance Configuration page provides basic information about the instance such as instance name and display name, description, the URL of the physical MSAS instance to which this logical instance is bound, the number of URLs and applications in the instance, and version data. You can modify the display name and the description for the instance. It also provides version information for the configuration.

To view general information about the instance:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

   c. In the MSAS Environments page, click **MSAS** or **Instances** in the MSAS tile. The MSAS Instances Summary page opens in a new tab.

      The first 8 instance in the environment are displayed. Click **Show More** to show additional instances.

   d. If necessary, use the Search field to refine the list of instances or to locate a specific instance. Enter all or part of a name in the Search field and press the search icon

   e. In the MSAS Instances Summary page, click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. In the MSAS Instance Configuration page, click the **General** tab to view the basic information about the instance.

Field	Description
Name	MSAS Instance ID. This field is read-only.
Display Name	Meaningful name used to identify the instance. This field is editable.
Description	Description of the instance. This field is editable.
MSAS URLs	Host and port of the physical MSAS instances to which this logical instance is bound. A logical MSAS instance can be bound to more than one physical instance.
Stats	Number of URLs and Applications configured in the instance.
Version Information	Version number of the instance, including the last user to update it and when it was updated.

6.2.2 Configuring the Identity Store Profile

The **Identity Store Profiles** tab of the MSAS Instance Configuration page provides the ability to add an identity store profile to the MSAS instance, edit an existing profile. delete a profile, and set the default.

An identity store profile is a logical representation of a user repository. All user and group entities are present in this identity store. There can be multiple profiles associated with an MSAS instance, and one profile can be marked as the default against which all authentication and user profile queries will occur.

> **Note:** You can also configure identity store profiles using WLST as described in "Configuring an Identity Store Profile Using WLST" on page 6-31.

The identity store configuration is stored in an Identity Profile Document in the MSAS Repository.

To configure an identity store profile:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

   c. In the MSAS Environments page, click **MSAS** or **Instances** in the MSAS tile. The MSAS Instances Summary page opens in a new tab.

      The first 8 instance in the environment are displayed. Click **Show More** to show additional instances.

   d. If necessary, use the Search field to refine the list of instances or to locate a specific instance. Enter all or part of a name in the Search field and press the search icon

   e. In the MSAS Instances Summary page, click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

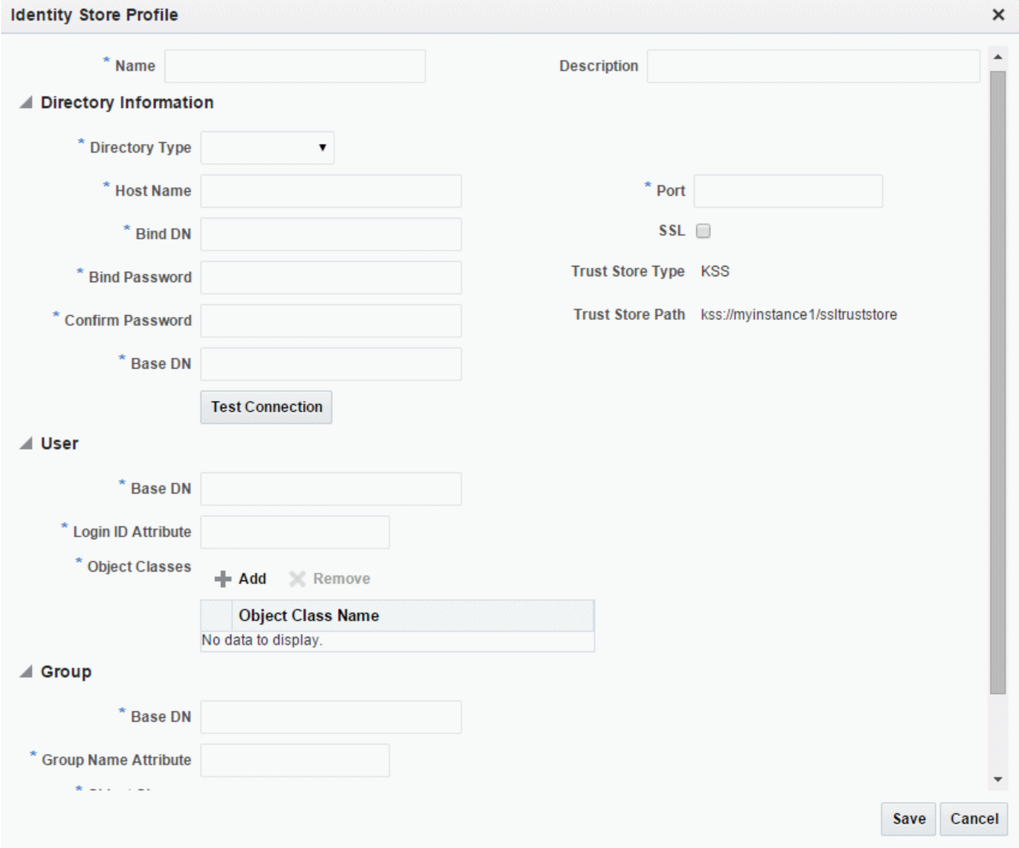
2. In the MSAS Instance Configuration page, click the **Identity Store Profiles** tab.

3. To add an identity profile, click **Add**.

   To edit an existing identity profile, click **Edit**. Note that if you are editing an existing profile, the fields are prepopulated with the configuration information for the existing profile.

   The Identity Store Profile page displays, as shown in Figure 6–1

Figure 6–1 Identity Store Profile Page



Trust Store Type KSS

Trust Store Path kss://myinstance1/ssltruststore

4. In the Identity Store Profile page, enter a meaningful name and description in the **Name** and **Description** fields.

5. In the Directory Information section, set the directory type, host name, and credential details for the identity store. When you have completed all the fields, click **Test Connection** to test the connection to the directory.

   If the connection to the identity store fails, ensure that the values you provided are correct. If the identity store is configured on an SSL port, verify that the SSL truststore contains the trusted SSL certificate. If necessary, you can import this trusted certificate as described in "Configuring the SSL Keystore and Truststore" on page 6-25.

Field	Description
Directory Type	Select one of the following directory types from the menu: ■ Active Directory ■ OID (Oracle Internet Directory) ■ ODSEE (Oracle Directory Server Enterprise Edition) ■ OUD (Oracle Unified Directory) ■ WLS-LDAP (Embedded LDAP in WebLogic Server)
Host Name	Host name of the server running the selected directory.
Port	Port used to access the selected directory.
SSL	If connecting using an SSL port, select this control to enable SSL.

Field	Description
Trust Store Type	For Mobile Security Access Server, the supported trust store type is KSS. This field is read only
Trust Store Path	Fully qualified path to the trust store. By default, this path is `kss://`*`msas_id`*`/ssltruststore`, where *`msas_id`* is the MSAS ID of the instance with which this identity store profile is associated. This field is read only.
Bind DN	DistinguishedName (DN) of the user to connect to the directory.
Bind Password/Confirm Password	Bind password for the Bind DN to connect to the selected directory.
Base DN	LDAP Searchbase under which all users and groups are located in the LDAP directory. For example, `cn=us, dn=mycompany, dc=com`

6. In the User section, set the user names, base DN, and object classes for the identity store profile.

Field	Description
Base DN	Container under which the users exist. For example, `cn=users,dn=mycompany,dc=com`.
Login ID Attribute	Login ID for the user. Typically this is `uid` or `mail` attribute in the LDAP. In Active Directory, this refers to the `UserPrincipalName`.
Object Classes	Fully qualified names of the schema classes used to represent users. By default, this field is set to the standard LDAP objectclass `inetOrgPerson`. To add an object class, click **Add**, then enter the value in the Object Class Name field. To remove an object class, select the row containing the class to be removed then click **Remove**.

7. In the Group section, set the group names, base DN, and object classes for the identity store profile.

Field	Description
Base DN	Searchbase for the group entries in the LDAP directory. For example, `cn=group,dn=mycompany,dc=com`.
Group Name Attribute	Attribute that uniquely identifies the name of the group or role. For example, `cn`.
Object Classes	Fully qualified names of the schema classes used to represent groups. By default, this refers to the LDAP standard objectclass of `groupofuniquenames`. In Active Directory this is `group`. To add an object class, click **Add**, then enter the value in the Object Class Name field. To remove an object class, select the row containing the class to be removed then click **Remove**.

8. When you have completed all of the fields, click **Save**.

9. If you have configured more than one identity store profile, select the profile to use as the default. To do so, select the profile in the table and click **Set as default**.

When set, all authentication and user profile queries will occur against the default identity store profile.

> **Note:** When you configure the identity store for an MSAS instance using the `idmConfig` tool with the `-configOMSS mode=OMSAS` arguments, the identity store that is created automatically becomes the default. To use an identity profile created as described in this section as the default profile, you must set it as described in this step.

10. Click **Apply**.
11. Restart the MSAS server.

6.2.3 Configuring Trusted Issuers and DN Lists for Signing Certificates

The **Authentication** tab of the MSAS Instance Configuration page enables you to define SAML and JWT trusted issuers and a list of trusted distinguished names (DNs) for SAML and JWT signing certificates.

> **Note:** You can also define SAML and JWT trusted issuers using WLST, as described in "Defining Trusted Issuers and Managing DN Lists Using WLST" on page 6-39.

The list of trusted issuers that you define here becomes the default list that is applicable to all applications in the instance. When you add an issuer using this method, it does not require a server restart.

By default, MSAS checks the incoming issuer name against the list of configured issuers, and checks the SAML and JWT signatures against the configured certificates in the MSAS keystore. If you defined a trusted DNs list for a trusted issuer, MSAS also verifies that the SAML and JWT signatures are signed by a certificate whose DN belongs to the trusted DN list.

Configuration of the trusted DNs list is optional; it is available for users that require more fine-grained control to associate each issuer with a list of one or more signing certificates. If you do not define a list of DNs for a trusted issuer, then MSAS allows signing by any certificate, as long as all the intermediate certificates and the CA certificate in the certificate chain for the signing certificate are present in the MSAS keystore. If the signing certificate is self-signed, it must be in the keystore itself.

Important Notes:

- Using the SAML and JWT Trust tables on the Authentication tab, you define the DNs of the *signing certificates*, not the certificates themselves.
- The CA and intermediate certificates for the signing certificate must be in the MSAS keystore regardless of whether the signing certificate is in the keystore or passed in the message.
- For two-way SSL:
  - The certificate needs to be imported into the trust store for the Java EE container.
  - The DN of the client SSL certificates are used for validation and must be present in the trusted DNs list.

- In all cases, the signing certificates must be trusted by the certificates present in the MSAS keystore.

Use the following procedure to add SAML or JWT issuers, define a trusted DN list for SAML or JWT signing certificates, or to add a DN to the DN list for an issuer:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the Mobile Security tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. In the MSAS Instance Configuration page, select the **Authentication** tab.

3. In the SAML Trust or JWT Trust section of the page, shown in Figure 6–2, do one of the following:

   - To add a SAML trusted issuer and define a trusted DN list for trusted STS servers, click **Add** in the Trusted STS table. Use this list for SAML HOK and SAML bearer.

   - To add a trusted issuer and define a trusted DN list for trusted SAML clients, click **Add** in the Trusted Clients table. Use this list for SAML sender vouches.

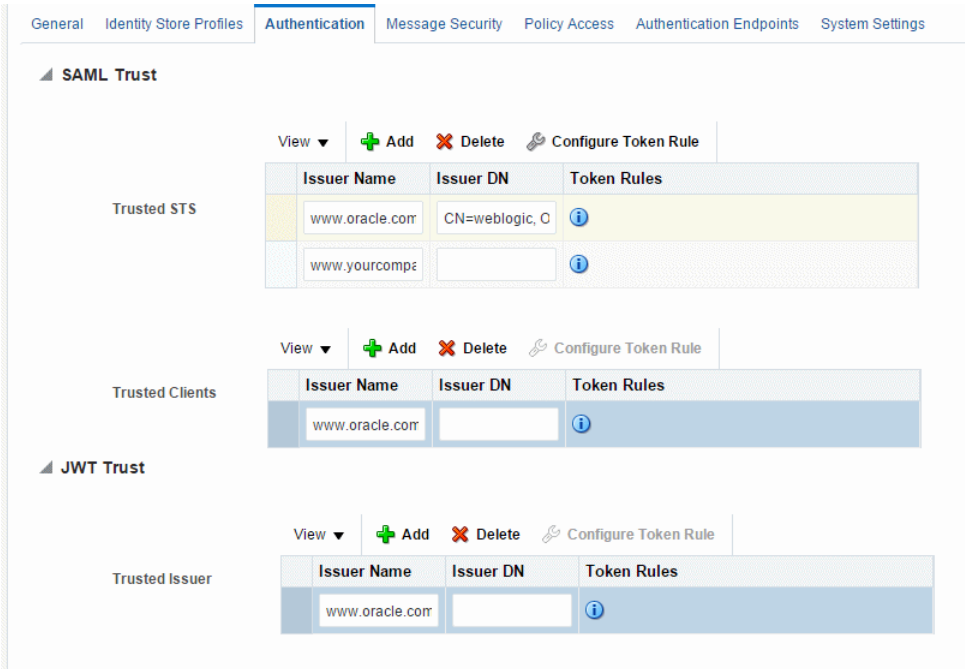
   - To add a trusted issuer and define a trusted DN list for trusted JWT issuers, click **Add** in the Trusted Issuer table.

4. In the **Issuer Name** column, enter a trusted issuer, for example `www.yourcompany.com`. The default value for the predefined SAML and JWT client policies is `www.oracle.com`.

5. In the **Issuer DN** column, enter the DN list for the trusted issuer. Use a string that conforms to RFC 2253. For example, the trusted DN for the trusted issuer `www.oracle.com` is `CN=weblogic, OU=Orakey Test Encryption Purposes Only, O=Oracle, C=US`. To add more than one DN for an issuer, separate each DN with a semicolon (`;`).

   For more information about RFC 2253, see `http://www.ietf.org/rfc/rfc2253.txt`.

Figure 6–2 SAML and JWT Trust Section of Authentication Tab



SAML Trust

Trusted STS

Issuer Name	Issuer DN	Token Rules
www.oracle.com	CN=weblogic, O	ⓘ
www.yourcompa		ⓘ

Trusted Clients

Issuer Name	Issuer DN	Token Rules
www.oracle.com		ⓘ

JWT Trust

Trusted Issuer

Issuer Name	Issuer DN	Token Rules
www.oracle.com		ⓘ

6. Optionally, add additional trusted issuers and DN lists. To do so, repeat steps 3 through 6.
7. To add a DN to the DN list for a trusted issuer, select the trusted issuer in the table and append the DN to the list of DNs in the **Issuer DN** column. Be sure to separate each DN by a semicolon (;).
8. To delete a trusted issuer, DN list, or individual DN from the list:
   - To delete a trusted issuer and the DN list, select the row containing the issuer to be deleted and click **Delete**.
   - To delete a DN list, clear the **Issuer DN** field for the DN to be deleted. Note that any configured token rules are also deleted.
   - To delete a DN from the DN list, select the row for the trusted issuer and delete the DN from the list.

   Note: The Configure Token Rule feature is reserved for future use.

9. When you have finished configuring the necessary issuers and DN lists, click **Apply**.
10. Optionally, if you want these changes to go into effect immediately, you can use the synchronize feature as described in "Synchronizing MSAS Instance Configuration" on page 2-2.

6.2.4 Configuring Message Security

The **Message Security** tab of the MSAS Instance Configuration page enables you to specify signature and encryption aliases for the MSAS keystore, and to configure message security settings such as clock skew and message expiration time. This configuration is described in the following sections:

- Configuring the MSAS Keystore
- Configuring Security Settings

> **Note:** You can also configure message security using WLST, as described in "Configuring Message Security Using WLST" on page 6-46.

6.2.4.1 Configuring the MSAS Keystore

When you create an MSAS instance using either the MSAS console or the `configMSAS` script, the MSAS signature keystore is created by default, using the MSAS instance name as the stripe name, such as `kss://myinstance/keystore`. The signing certificate of the MSM server using the alias `msm_sign_cert0` is automatically imported into the keystore. If the signing certificate for the MSM is a certificate chain, the additional certificates are imported as well, for example `msm_sign_cert1`, `msm_sign_cert2`, and so on.

When you configure the identity store for the instance, it imports the default signature and encryption keys using the alias *`msas_id_`*`orakey`. Because this KSS keystore is defined at the instance level, the signature and encryption keys apply to all applications in the instance.

> **Note:** The MSAS keystore is not used for the SSL keys. For details about the SSL keystore and truststore, see Chapter 7, "Configuring the SSL Keystore and Truststore."

You can use the fields in the Keystore section to generate a keypair and import keys into the keystore, and to delete keys from the keystore. To do so:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. In the MSAS Instance Configuration page, click the **Message Security** tab.

3. Click the **Sign Alias** alias key to import a new private signature key, or to generate a keypair.

   If you are configuring a logical instance that has not yet been bound to a physical instance, click **Click to Add**.

4. To import a signature key and alias from a keystore, click **Import From Keystore**.

5. Click **Choose File** to select the key file to be imported from the file system, and enter the alias of the key to be imported in the **Alias** field. If the keystore from

which the key is being imported is password protected, complete the **Keystore Password** and **Alias Password** fields.

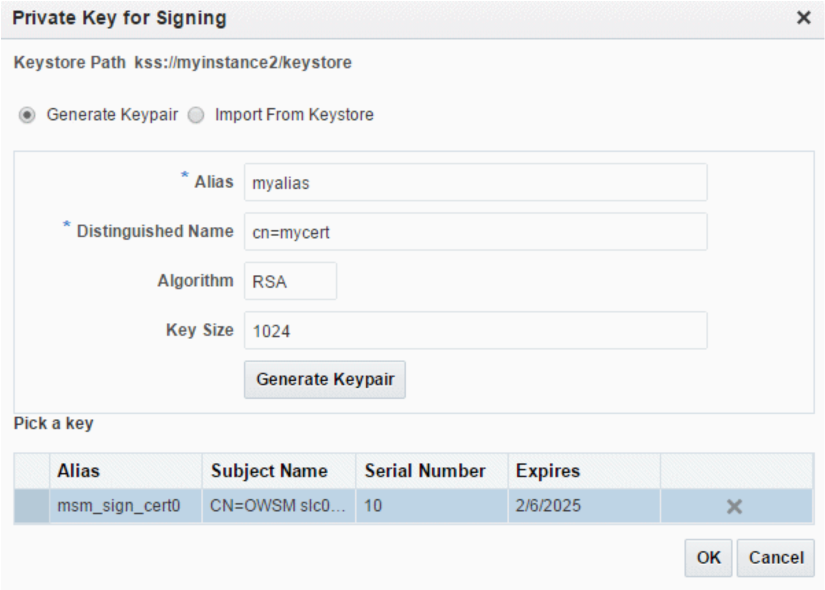
6. Click **Import**.

   The key is added to the list of keys in the table.

7. To generate a new keypair, click **Generate Keypair**, provide an alias and a distinguished name for the keypair as shown in Figure 6–3.

Field	Description
Alias	Enter an alias for the keypair. This field is required.
Distinguished Name	Specify the distinguished name (DN) of the certificate wrapping the keypair. This field is required.
Algorithm	Symmetric key algorithm. The default is RSA.
Keysize	The RSA keysize. The default is 1024 bytes.

Figure 6–3 Generate Keypair Options in Private Key for Signing Window



Private Key for Signing

Keystore Path kss://myinstance2/keystore

● Generate Keypair ○ Import From Keystore

* Alias: myalias

* Distinguished Name: cn=mycert

Algorithm: RSA

Key Size: 1024

Pick a key

Alias	Subject Name	Serial Number	Expires	
msm_sign_cert0	CN=OWSM slc0...	10	2/6/2025	✕

8. Click **Generate Keypair**.

   The keypair is added to the table.

9. Select the key in the table to be used as the signature key and click **OK**.

10. To import a new private key used for decrypting messages, or to generate an encryption keypair, click the **Encrypt Alias** alias key, then repeat steps 4 through 8 as required.

11. Select the alias in the table to be used as the decryption key for the instance and click **OK**.

12. To delete a signature or encryption key from list of available keys, click the Sign Alias or Encrypt Alias key and in the pop-up window, select the key to be deleted and click **X**, click **Yes** when prompted, then click **OK**.

    The entry is deleted from the table.

13. Click **Apply**.

6.2.4.2 Configuring Security Settings

The Security Settings section of the **Message Security** tab allows you to tune security settings such as the default message timestamp skews between system clocks, and the message expiration time.

To configure the security settings:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **Message Security** tab.

3. In the Security Settings section of the page, set the following properties as required for your environment, then click **Apply**:

   - **Clock Skew** – Specify the tolerance of time differences, in milliseconds, between client and server machines. For example, when timestamps are sent in a message to a service that follows a different time zone, this property allows for the specified time tolerance. The default value is 360,000 milliseconds (6 minutes). To adjust the clock skew, enter a new value in the field or click the up/down arrows to increase or decrease the default value.

     You should configure this property as follows:

     – Increase the clock skew when the client and service are running on different systems and their system clocks are not in sync, which could result in the service rejecting messages from the client, with an error indicating the timestamp validation failed. Increasing the clock skew accounts for the difference in clocks between the client and the web service. For example, if the difference between the service clock and the client clock is 10 minutes, increase the Clock Skew on the system hosting the service to 10 minutes (600000 milliseconds).

     – Decrease the clock skew if you want to narrow the window in which the service is willing to accept messages from clients to avoid replay attacks.

   - **Client Clock Skew**—Tolerance of time, in seconds, that is used to calculate the NotBefore and NotOnOrAfter conditions for SAML and JWT token generation. Together, these conditions define the lower and upper boundaries to limit the validity of the token.

   - **Message Expiration Time**— Specify the duration of time, in seconds, before a message expires after its creation. This property is used in cases where a timestamp is sent across in the token to verify if the timestamp has expired or not. It is also used to control the timestamp expiry window on the client side when the message is created. The value specified here applies to all message protection and SAML and JWT assertion timestamp elements. The default value is 300,000 milliseconds (5 minutes).

If the message expires when received by the service even when there is no time difference between the client's and service's clocks, then the message expiration time must be increased. The message expiration time is derived from the values of Message Expiration Time and the expiry time in the incoming message, and is the lesser of the two.

For example, if the server's Message Expiration Time is set to 5 minutes and the incoming message expiry time is 6 minutes, then the effective timestamp validation window is only 5 minutes and the incoming message is only be valid in that 5 minute window. In this case, you need to increase the Message Expiration Time at the service side. (Increasing the timestamp expiry on the incoming message will not fix the problem because the message expiration time is derived from both values and is the lesser of the two.)

On the other hand, if the server's Message Expiration Time is 5 minutes and the incoming message expiry time is 3 minutes, then the expiry time in the incoming message (that is, at the client side) must be increased.

Note: A higher value of the Message Expiration Time may lead to a security vulnerability.

6.2.5 Configuring the Cache Refresh Time

The **Policy Access** tab of the MSAS Instance Configuration page enables you to configure the amount of time to wait between cache refreshes of the MSAS run-time cache. The physical MSAS instance fetches the configuration for the logical MSAS instance maintained in the MSAS repository and caches it. This cache is refreshed periodically based upon the setting defined in this section.

Note: You can also configure the cache refresh time using WLST, as described in "Configuring the Cache Refresh Time Using WLST" on page 6-49.

To configure the cache refresh time:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **Policy Access** tab.

3. In the Cache Management section of the page, configure the **Cache Refresh Time** property, which is the number of milliseconds to wait between cache refreshes. To set this property, enter a new value in the field, or click the up/down arrows to

increase or decrease the default value. The default is 86,400,000 milliseconds (24 hours).

> **Note:** The remaining fields on this tab are reserved for future use.

4. Click **Apply** to save the property updates.

6.2.6 Configuring Authentication Endpoints

The **Authentication Endpoints** tab of the MSAS Instance Configuration page enables you to configure the endpoints used for initial authentication of the user for each of the initial authentication types: KINIT, PKINIT, OAuth2, and Oracle Access Manager Mobile and Social.

> **Note:** You can also configure the authentication endpoints using WLST, as described in "Configuring the Authentication Endpoints Using WLST" on page 6-50.

For initial authentication and generation of the session token, Mobile Security Access Server provides a virtual (reverse proxy) application that contains a URL for each of the initial authentication types. Each of these URLs is provisioned with a predefined policy for authentication on-request, and a session token (SToken) generation policy on response.

This topic contains the following sections:

- Configuring KINIT and PKINIT Authentication
- Configuring OAuth2 Confidential Client Authentication
- Configuring Oracle Access Manager Mobile and Social (OAMMS) Authentication
- Configuring the Crypto Service

6.2.6.1 Configuring KINIT and PKINIT Authentication

The KINIT & PKINIT section of the **Authentication Endpoints** tab enables you to configure the properties in the Kerberos configuration file `krb5.conf` that are required for Kerberos Password Authentication (KINIT) and Public Key Cryptography for Initial Authentication (PKINIT) to work.

The fields in this section allow you to add, edit, and delete Kerberos realms, add and delete DNS domains, specify Kerberos encryption types, specify the logging location for the Kerberos and KCM messages, and enable the use of PKINIT trust anchors. Note that manual changes to this `krb5.conf` file are not persisted because it is overwritten each time MSAS is restarted.

For more advanced configuration, you should create a `krb5.conf` manually and save it to *`instance_root/instance_name`*`/config/default` directory. Once you do so, it takes precedence over the `krb5.conf` file that is created using the console. You can then customize it, for example, to point to specific domain controllers or to accommodate environments with alternate UPN suffixes. For more information, see "Advanced Kerberos Configuration" on page 6-57.

To configure KINIT and PKINIT:

1. Navigate to the MSAS Instance Configuration page:

a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

b. In the Mobile Security Access Server section, click **Environments**.

   The MSAS Environments page opens in a new tab.

c. Click **MSAS** or **Instances** in the MSAS tile.

   The MSAS Instances Summary page opens in a new tab.

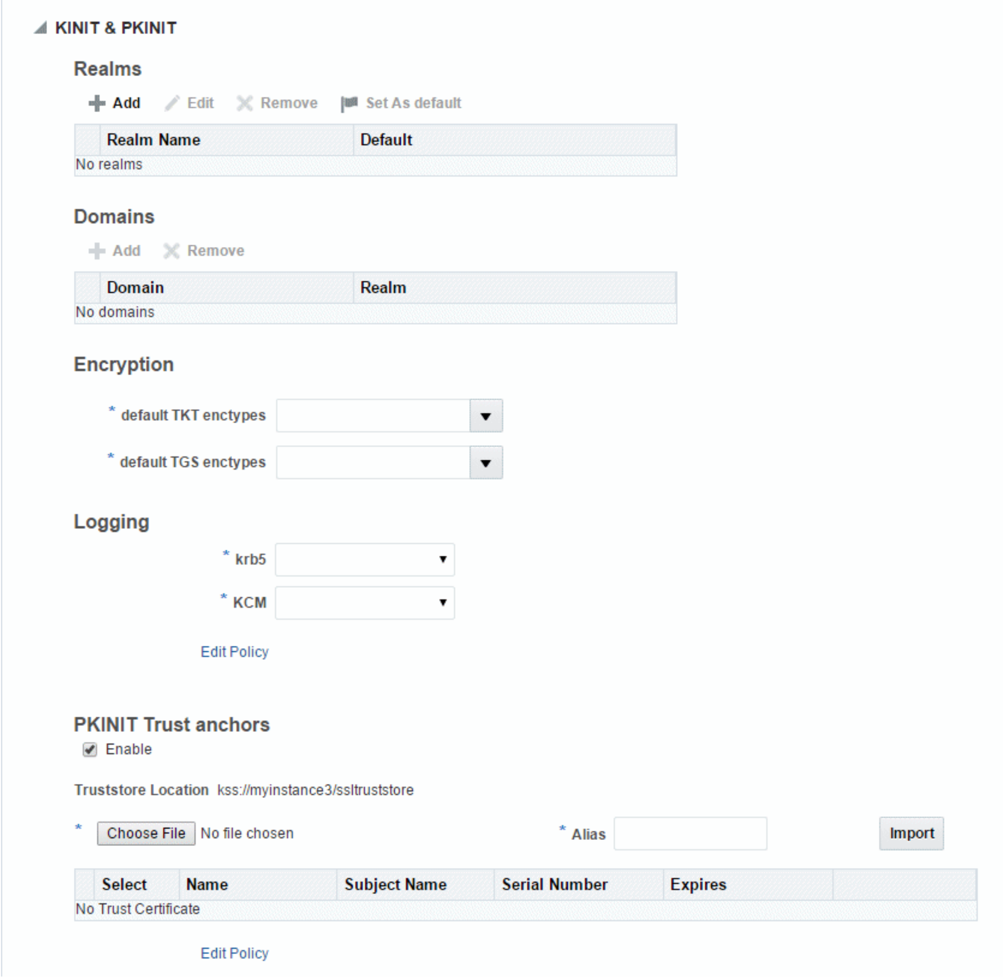
d. Click the instance name or **Configure** in the tile for the desired instance.

   The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **Authentication Endpoints** tab.

3. Expand the **KINIT & PKINIT** section (if it is not already expanded). The section is shown in Figure 6–4.

Figure 6–4 KINIT & PKINIT Section on Authentication Endpoints Tab



Realm Name	Default
No realms	

Domain	Realm
No domains	

☑ Enable

Truststore Location kss://myinstance3/ssltruststore

Select	Name	Subject Name	Serial Number	Expires
No Trust Certificate				

4. In the Realms section, configure the Kerberos realm.

   a. Click **Add** to add a new Kerberos realm.

   b. In the Realm window, provide a name, KDC host and port, and a default domain for the realm, then click **OK**.

Field	Description
Name	Enter a name for the realm. The realm name must match the REALM name defined during the Active Directory setup.
KDC host	Enter the host for the KDC server running the realm specified in the Name field.
KDC port	Optionally, enter the port of the KDC server running the realm specified in the Name field.
Default Domain	Enter the name of the default domain in the field, or select a default domain from the menu. Typically, the domain name is in lower case, for example `example.com`.

Note that the domain name is automatically added to the Domains table.

c. To set a default realm, select the realm in the Realms table and click **Set As Default**.

d. To edit an existing realm, select the realm in the Realms table and click **Edit**. Note that you cannot change the name of a realm. In the Realm window, enter a new KDC host, port, or default domain, then click **OK**.

e. To delete a realm, select the realm in the Realms table and click **Remove**.

5. In the Domains section, configure the DNS domains.

   a. Click **Add** to add a DNS domain.

   b. In the **Domain** field, enter the name, then in the **Realm** field, select an associated realm from the menu.

   c. To delete a domain, select the domain in the Domains table and click **Remove**.

6. In the Encryption section, select the TKT and TGS encryption types.

   a. From the **default TKT enctypes** menu, select the session key encryption type that the client should use when making an initial authentication request (AS-REQ).

   b. From the **default TGS enctypes** menu, select the session key encryption type that the client should use when requesting a service ticket from the TGS (TGS_REQ).

   For a list of the supported encryption types, refer to the MIT Kerberos Documentation, "Encryption types" at `http://web.mit.edu/Kerberos/krb5-1.12/doc/admin/conf_files/kdc_conf.html#encryption-types`.

7. In the Logging section, specify the logging location for the Kerberos and Kerberos Cache Manager (KCM) messages.

   Select the log locations to use for the Kerberos configuration messages (krb5 menu, and the KCM messages (KCM menu). Select **STDERR** to log messages using the standard error stream. Select **File** to log messages to a file, and in the empty field, enter the log file location.

8. To view or edit the policy configuration of the KINIT endpoint:

   a. Click **Edit Policy** to display the URL Policy Configuration page for the endpoint in a new tab named **AuthnService**.

   Two policies are attached by default:

The *HTTP Kerberos Password Authentication Service Policy* is attached to the On-Request endpoint. This internal policy enables the Kerberos password authentication.

The *HTTP Session Token Issue Policy* is attached to the On-Response endpoint. This policy issues a session token with the authenticated user ID to the client.

b. Select the policy name, or click the **Options** menu icon ☰ then **Edit**, to view the policy details and configure policy overrides.

c. Optionally, click the **Overrides** tab to configure property overrides.

In the *HTTP Kerberos Password Authentication Service Policy* you can configure the `keystore.sig.csf.key`, which is the CSF key used for the signature key name and password.

In the *HTTP Session Token Issue Policy*, you can configure the `keystore.sig.csf.key` and the `keystore.enc.csf.key`, which is the alias of the certificate to be used for encrypting the session token.

If you want to use a different signature or encryption key than the one configured for the instance, you can do so using these configuration overrides. To add a signature or encryption CSF key to the keystore press **Click to add**, then click **Generate Keypair** to generate a keypair or **Import from Keystore** to import a key from a keystore. When you have added the desired key, select the key in the table and click **OK**.

> **Note:** To ensure proper authentication on this endpoint, you should not delete the default policy attachments, or attach additional policies.

d. If you have made changes to the policy configuration, click **Validate** to validate the policy, then click **Apply** to save your changes. Close the **AuthnService** tab.

If you have not made any changes to the endpoint configuration, you can close the tab to return to the MSAS Instance Configuration page.

9. In the PKINIT Trust Anchors section, you can enable trust anchors for PKINIT authentication. PKINIT trust anchors are stored in the trust store. You can also view and edit the policy configuration of the PKINIT endpoint.

a. Click **Enable** to enable the use of PKINIT anchors to trust the authority issuing the KDC certificate,

b. If desired, import a certificate into the truststore. To do so, click **Choose File** to select a file from the file system, enter an alias for the certificate in the **Alias** field, and click **Import**. The certificate is added to the table.

c. Select the trusted certificate from the table to be used as the PKINIT anchor. The certificate you select must be the first certificate in the certificate chain. MSAS will automatically fetch the complete chain using the selected certificate as the starting point.

d. To view or edit the policy configuration of the PKINIT endpoint, click **Edit Policy** to display the URL Policy Configuration page for the endpoint in a new tab named **AuthnService**.

Two policies are attached by default:

The *HTTP Kerberos PKI Authentication Service Policy* is attached to the On-Request endpoint. This internal policy enables the Kerberos PKI authentication.

The *HTTP Session Token Issue Policy* is attached to the On-Response endpoint. This policy issues a session token with the authenticated user ID to the client.

e. Select the policy name, or click the **Options** menu icon ☰ then **Edit**, to view the policy details and configure policy overrides.

f. Optionally, click the **Overrides** tab to configure property overrides.

In the *HTTP Kerberos PKI Authentication Service Policy* you can configure the `keystore.sig.csf.key`, which is the CSF key used for the signature key name and password.

In the *HTTP Session Token Issue Policy*, you can configure the `keystore.sig.csf.key` and the `keystore.enc.csf.key`, which is the alias of the certificate to be used for encrypting the session token.

If you want to use a different signature or encryption key than the one configured for the instance, you can do so using these configuration overrides. To add a signature or encryption csf key to the keystore press **Click to add**, then click **Generate Keypair** to generate a keypair or **Import from Keystore** to import a key from a keystore. When you have added the desired key, select the key in the table and click **OK**.

> **Note:** To ensure proper authentication on this endpoint, you should not delete the default policy attachments, or attach additional policies.

g. If you have made changes to the policy configuration, click **Validate** to validate the policy then click **Apply** to save your changes. Close the **AuthnService** tab.

If you have not made any changes to the endpoint configuration, you can close the tab to return to the MSAS Instance Configuration page.

10. Click **Apply** at the top of the MSAS Instance Configuration page to save your changes.

6.2.6.2 Configuring OAuth2 Confidential Client Authentication

In order for the Mobile Security Access Server to authenticate users against Oracle Access Manager and retrieve Oracle Access Manager and OAuth tokens for integrated single sign on, the Mobile Security Access Server must be registered as an OAuth Confidential Client with the Oracle Access Manager OAuth Service.

The OAuth2 Confidential Client section of the **Authentication Endpoints** tab enables you to configure the OAuth2 confidential client endpoint required, and to specify the client ID and secret in the Credential Store Framework (CSF).

To configure the OAuth2 confidential client endpoint:

1. Navigate to the MSAS Instance Configuration page:

a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

b. In the Mobile Security Access Server section, click **Environments**.

The MSAS Environments page opens in a new tab.

c. Click **MSAS** or **Instances** in the MSAS tile.

   The MSAS Instances Summary page opens in a new tab.

d. Click the instance name or **Configure** in the tile for the desired instance.

   The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **Authentication Endpoints** tab.

3. Expand the **OAuth2 Confidential Client** section (if it is not already expanded).

4. If it is not already configured, in the **Endpoint** field add the host and port for the OAuth Service Profile Endpoint to the endpoint URL, for example `http://myhost:port/ms_oauth/oauth2/endpoints/oauthservice`. This is the endpoint to which the MSAS server creates JWT User Token and OAM Tokens for OAuth2 Confidential Client Authentication flow.

5. Click **Edit Policy** to view the policy configuration for the endpoint and to specify the CSF client key. The URL Policy Configuration page for the endpoint opens in a new tab named **AuthnService**.

   Two policies are attached by default.

   The *HTTP OAuth2 Confidential Client Over SSL Policy* is attached to the On-Request policy enforcement endpoint. This internal policy performs OAuth2 confidential client authentication and creates OAuth and OAM tokens.

   The *HTTP Session Token Issue Policy* is attached to the On-Response policy enforcement endpoint. This policy issues a session token with the authenticated user ID to the client.

6. Select the *HTTP OAuth2 Confidential Client Over SSL Policy* policy name, or click the **Options** menu icon ≡ then **Edit**, to view the policy details and configure policy overrides.

7. Optionally, click the **Overrides** tab to configure the policy overrides.

   - In the *HTTP OAuth2 Confidential Client Over SSL Policy*, you can configure the `oauth2.client.csf.key` property. The default value of this property in the policy is set to `oauth2.confidential.client.credentials`. When you create the instance using `configMSAS`, it creates a CSF key for this property with a default client ID and password. The client ID is in the format *`MSAS_Instance_ID`*`_MSASClient`, for example `myinstance1_MSASClient`, and the password is randomly generated.

     To change this value, enter the OAuth2 client CSF key in the **Value** field then click **Apply**.

   - In the *HTTP Session Token Issue Policy*, you can configure the `keystore.sig.csf.key` and the `keystore.enc.csf.key`, which is the alias of the certificate to be used for encrypting the session token.

     If you want to use a different signature or encryption key than the one configured for the instance, you can do so using these configuration overrides. To add a signature or encryption csf key to the keystore press **Click to add**, then click **Generate Keypair** to generate a keypair or **Import from Keystore** to import a key from a keystore. When you have added the desired key, select the key in the table and click **OK**.

Note: To ensure proper authentication on this endpoint, you should not delete the default policy attachments, or attach additional policies.

8. If you have made changes to the policy configuration, click **Validate** to validate the policy then click **Apply** to save your changes. Close the **AuthnService** tab.

   If you have not made any changes to the endpoint configuration, you can close the tab to return to the MSAS Instance Configuration page.

9. Click **Apply** at the top of the MSAS Instance Configuration page to save your changes.

6.2.6.3 Configuring Oracle Access Manager Mobile and Social (OAMMS) Authentication

The OAuth2 Mobile Client section of the **Authentication Endpoints** tab enables you to configure the OAuth2 mobile client endpoint required for Oracle Access Manager Mobile and Social mobile client authentication, and to specify the client ID and secret in the Credential Store Framework (CSF).

To configure the OAuth2 mobile client endpoint:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **Authentication Endpoints** tab.

3. Expand the **OAuth2 Mobile Client** section (if it is not already expanded).

4. If it is not already configured, in the **Endpoint** field add the host and port for the OAuth Service Profile Endpoint to the endpoint URL, for example `http://myhost:port/ms_oauth/oauth2/endpoints/oauthservice`. This is the endpoint to which the MSAS server creates JWT User Token and OAM Tokens for OAuth2 Mobile Client Authentication flow.

5. Click **Edit Policy** to view the policy configuration for the endpoint and to specify the CSF client key. The URL Policy Configuration page for the endpoint opens in a new tab named **AuthnService**.

   Two policies are attached by default.

   The *HTTP OAuth2 Mobile Client Over SSL Policy* is attached to the On-Request policy enforcement point. This internal policy performs OAuth2 mobile client authentication and creates OAuth and OAM tokens.

   The *HTTP Session Token Issue Policy* is attached to the On-Response policy enforcement point. This policy issues a session token with the authenticated user ID to the client.

6. Select the *HTTP OAuth2 Mobile Client Over SSL Policy* policy name, or click the **Options** menu icon ☰ then **Edit**, to view the policy details and configure policy overrides.

7. Optionally, click the **Overrides** tab to configure the policy overrides.

   - In the *HTTP OAuth2 Mobile Client Over SSL Policy*, you can configure the `oauth2.mobile.client.csf.key` property. The default value of this property in the policy is set to `oauth2.mobile.client.id`. When you create the instance using `configMSAS`, it creates a CSF key for this property with a default client ID and password. The client ID is in the format *`MSAS_Instance_ID_`*`OracleContainer`, for example `myinstance1_OracleContainer`. The password can be any value as it is not used in this configuration.

     To change this value, enter the OAuth2 client CSF key in the **Value** field then click **Apply**.

   - In the *HTTP Session Token Issue Policy*, you can configure the `keystore.sig.csf.key` and the `keystore.enc.csf.key`, which is the alias of the certificate to be used for encrypting the session token.

     If you want to use a different signature or encryption key than the one configured for the instance, you can do so using these configuration overrides. To add a signature or encryption csf key to the keystore press **Click to add**, then click **Generate Keypair** to generate a keypair or **Import from Keystore** to import a key from a keystore. When you have added the desired key, select the key in the table and click **OK**.

     Note: To ensure proper authentication on this endpoint, you should not delete the default policy attachments, or attach additional policies.

8. If you have made changes to the policy configuration, click **Validate** to validate the policy then click **Apply** to save your changes. Close the **AuthnService** tab.

   If you have not made any changes to the endpoint configuration, you can close the tab to return to the MSAS Instance Configuration page.

9. Click **Apply** at the top of the MSAS Instance Configuration page to save your changes.

6.2.6.4 Configuring the Crypto Service

The Crypto Service section of the **Authentication Endpoints** tab enables you to configure the key rollover feature in the PKI Crypto Service and to modify the policy endpoint configuration if desired.

To configure the Crypto service endpoint:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **Authentication Endpoints** tab.
3. Expand the **Crypto Service** section (if it is not already expanded).
4. Optionally, you can enable the key rollover aliases feature. By default, the **Key Rollover Aliases** field is blank, and key rollover support is not enabled. To enable key rollover, select an alias from the menu, or select **All**. The aliases selected will be added to the ordered list of aliases used in the rollover feature. A decryption error will not be returned by the crypto service unless decryption of a given ciphertext fails with all archived private keys.

   Whenever you change the encryption/signing key of MSAS, be sure to add the new signing keys as a new entry and keep the existing keys in the keystore. The alias that you select here should be the older key alias. Be sure to do this every time that you add an encryption key to the keystore, always ensuring you are adding the older key to this list. Doing so ensures that data encrypted using older keys can still be decrypted even if you have changed the keys.

5. Click **Edit Policy** to view the policy configuration for the endpoint. The URL Policy Configuration page for the endpoint opens in a new tab named **AuthnService**.

   Two policies are attached by default to the On-Request policy enforcement point.

   - The *HTTP Session Token Verify Policy* verifies the session token, including the timestamp and signature, decrypts the encrypted data, and asserts the identity using the user ID from the session token. The request is rejected if the verification fails.
   - The *HTTP Action Security Policy* performs SKEK encryption and decryption.

6. Optionally, select the *HTTP Session Token Verify Policy* name or click the **Options** menu icon ☰ then **Edit**, to view the policy details and configure policy overrides.

   If you want to use a different signature or encryption key than the one configured for the instance, you can do so using these configuration overrides. You can configure the `keystore.sig.csf.key`, which is the CSF key used for the signature key name and password and the `keystore.enc.csf.key`, which is the alias of the certificate to be used for encrypting the session token.

   To add a signature or encryption csf key to the keystore, press **Click to add**, then click **Generate Keypair** to generate a keypair or **Import from Keystore** to import a key from a keystore. When you have added the desired key, select the key in the table and click **OK**.

   > **Note:** To ensure proper authentication on this endpoint, you should not delete the default policy attachments, or attach additional policies.

7. If you have made changes to the policy configuration, click **Validate** to validate the policy then click **Apply** to save your changes. Close the **AuthnService** tab.

   If you have not made any changes to the endpoint configuration, you can close the tab to return to the MSAS Instance Configuration page.

8. Click **Apply** at the top of the MSAS Instance Configuration page to save your changes.

6.2.7 Configuring System Settings

The **System Settings** tab of the MSAS Instance Configuration page enables you to configure settings for outbound message, proxy servers, load balancing, SSL, and logging. This configuration is described in the following sections:

- Configure Outbound Message Settings
- Configure Proxy Server Settings
- Configure Server Settings
- Configuring the SSL Keystore and Truststore

> **Notes:** For details on logging configuration, see Chapter 9, "Managing Log Files."
>
> You can also configure system settings using WLST as described in "Configuring System Settings Using WLST" on page 6-52.

6.2.7.1 Configure Outbound Message Settings

The Outbound Message Settings section on the **System Settings** tab enables you to configure the client connection to back-end services.

To configure the outbound message settings:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **System Settings** tab.
3. Expand the **Outbound Message Settings** section (if it is not already expanded).
4. Specify the connection pool settings and connection timeouts. To do, click in the field to see the default, and adjust the value by clicking the up and down arrows.

Field	Description
Total Connections in pool	Specify the maximum number of connections in a pool that a client can handle. The default is 512.
Maximum Connections per host	Specify the maximum number of connections in a pool, per host, that a client can handle. The default is 25.
Connection Timeout	Specify the maximum time in milliseconds a client can wait when connecting to a back-end host. The default is 20,000 ms.
Idle Connection pool Timeout	Specify the maximum time in milliseconds a client will keep idle connections in the pool.The default is 180,000 ms (3 minutes).

Field	Description
Request Timeout	Specify the maximum time in milliseconds a client can wait for a response. The default is 60,000 ms (1 minute).

5. Click **Apply** at the top of the page to save your changes.
6. Restart the MSAS server.

6.2.7.2 Configure Proxy Server Settings

The Proxy Server Settings section on the **System Settings** tab enables you to configure the proxy server used for outbound calls to the internet through MSAS for back-end applications and services.

To configure the proxy server settings:

1. Navigate to the MSAS Instance Configuration page:
   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.
   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.
   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.
2. Select the **System Settings** tab.
3. Expand the **Proxy Server Settings** section (if it is not already expanded).
4. Specify the settings for the proxy server, including the host name and port, user ID and password, and the list of hosts that will not use the proxy.

Field	Description
Name	Enter a name for the proxy server to uniquely identify it. This field is optional.
Host Name	Enter the host name of the proxy server.
Port	Enter the port number of the proxy server.
User Name	Enter the user ID to connect to the proxy server. **Note:** The User Name and Password is required only if the proxy server requires authentication.
Password	Enter the password corresponding to the proxy server user ID.
Hostnames without proxy	Enter one or more hosts that will not use the proxy server. This field supports the asterisk * wildcard, but only as a prefix and suffix. By default, this field contains the value `localhost, 127.0.0.1`.

5. Click **Apply** at the top of the page to save your changes.
6. Restart the MSAS server.

6.2.7.3 Configure Server Settings

The Server Settings section on the **System Settings** tab enables you to configure general server settings for the MSAS server, such as load balancing URLs, and Service Principal Name and URL mapping.

To configure the server settings:

1. Navigate to the MSAS Instance Configuration page:
   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.
   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.
   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.
2. Select the **System Settings** tab.
3. Expand the **Server Settings** section (if it is not already expanded).
4. Specify the load balancer URLs for the server.

Field	Description
Load Balancer URL	Enter a non-SSL URL for a front ending load balancer, for example `http://lbr.example.org:80`.
Load Balancer SSL URL	Enter an SSL URL for a front-ending load balancer, for example `https://lbr.example.org:443`.

5. In the Service Principal Name section, complete the fields to map a URL to a Service Principal Name. Service Principal Name is required for the NTLM and SPNEGO protocols.
   a. In the **URL** field, enter the Service Principal Name URL. You can use the `*` wildcard anywhere in the URL, for example `http*://example.host*80/* or *.example.org`.
   b. In the **Service Principal Name** field, enter the Service Principal Name in the form of *`SPN_SERVICECLASS/SPN_HOSTNAME`*.
   c. To add more Service Principal Names, click **Add** and, in the new row, enter the values in the **URL** and **Service Principal Name** field.
   d. To delete a Service Principal Name, select the row and click **Remove**.
6. Click **Apply** at the top of the page to save your changes.

6.2.7.4 Configuring the SSL Keystore and Truststore

The SSL Settings section on the **System Settings** tab enables you to add certificates to the SSL Truststore and private keys to the SSL keystore. The SSL keystore is used for inbound SSL connections to MSAS and as the MSAS identity keystore. The SSL truststore serves as the repository for trusted certificates used for trusting or authenticating client certificates (for two-way SSL).

To configure the SSL settings:

1. Navigate to the MSAS Instance Configuration page:

   a. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.

   b. In the Mobile Security Access Server section, click **Environments**.

      The MSAS Environments page opens in a new tab.

   c. Click **MSAS** or **Instances** in the MSAS tile.

      The MSAS Instances Summary page opens in a new tab.

   d. Click the instance name or **Configure** in the tile for the desired instance.

      The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

2. Select the **System Settings** tab.

3. Expand the **SSL Settings** section (if it is not already expanded).

   The **SSL TrustStore Location** and **SSL KeyStore Location** fields are read-only, and contain the location of the SSL truststore and SSL keystore for the instance. Note that only KSS keystores are supported, therefore these fields must contain a KSS URI.

4. To import a server certificate into the SSL trust store:

   a. Click **Click To Import**.

   b. In the Server Certificate window, click **Choose File** and select the certificate to be imported from the file system.

      Note: Only Base64-encoded certificates are supported.

   c. Enter an **Alias** for the certificate to be imported, select Trusted Certificate from the **Certificate Type** menu, and click **Import**.

      The imported certificate is added to the list of certificates in the table.

   d. Click **Close** to close the Server Certificate window.

5. To import a private key or to generate a keypair to add to the SSL keystore:

   a. Click **Click to add** to display the Private Key window.

      From the Private Key window you can generate a keypair, or import a private key from a JKS keystore.

   b. To generate a private keypair for the MSAS SSL identity key., click **Generate Keypair**.

      In the **Alias** field, provide an alias for the keypair.

      In the **Distinguished Name** field, enter the distinguished name of the certificate wrapping the keypair.

      In the **Algorithm** field, enter a symmetric key algorithm. The default is RSA.

      In the **Key Size** field, enter the RSA key size. The default is 1024 bytes.

      Click **Generate Keypair**. The keypair is created and added to the keypair table.

c. To import a Java keystore file into the keystore service, click **Choose File** and select the Java keystore file to be imported.

   If the JKS keystore from which the key file is being imported is password protected, enter the **Keystore Password** for the JKS keystore.

   Enter an **Alias** for the keypair, and, an **Alias Password** if the alias is password protected.

   Then click **Import**.

6. Click **Apply** at the top of the page to save your changes.
7. Restart the MSAS server.

6.3 Configuring an MSAS Instance Using WLST

You can use WLST commands to configure an MSAS instance, including identity store profiles, trusted issuers and DNs, message security, policy access, authentication endpoints, other security settings, and the MSAS heartbeat.

The majority of the properties used to configure an MSAS instance can be set using the `setMSASConfiguration` command. For details about using this command, see "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

Additional commands are provided to configure the identity store profile, trusted issuers and DN lists, the credential store, and the KSS keystore for MSAS. These commands are described in the relevant sections.

The following sections describe how to configure an MSAS instance using WLST commands.

- Accessing the MSAS WLST Commands
- Viewing MSAS Instance Configuration Using WLST
- Using the setMSASConfiguration WLST Command to Configure an MSAS Instance
- Configuring an Identity Store Profile Using WLST
- Defining Trusted Issuers and Managing DN Lists Using WLST
- Configuring Message Security Using WLST
- Configuring the Cache Refresh Time Using WLST
- Configuring the Authentication Endpoints Using WLST
- Configuring System Settings Using WLST
- Configuring the SToken Expiry Time
- Configuring the MSAS Heartbeat Using WLST
- Configuring Additional Server Settings Using WLST
- Configuring the Credential Store Using WLST

6.3.1 Accessing the MSAS WLST Commands

The WLST commands used to configure MSAS instances must be executed from the `ORACLE_IDM/common/bin` directory of your Mobile Security Manager (MSM) installation. Before running these commands, ensure that the Administration Server for the MSM domain is running.

To access the MSAS WLST commands:

1. Go to the `ORACLE_IDM/common/bin` directory of the Middleware home directory for your Mobile Security Manager installation, for example `/home/oracle/omsm/ORACLE_IDM/common/bin`.
2. Start WLST in offline mode using the following command:

   ```
   ./wlst.sh
   ```

   To use the MSAS WLST commands, you must use WLST in online mode.
3. Connect to the running Mobile Security Manager Administration Server using the `connect()` command.

   For example, the following command connects WLST to the Admin Server at the URL `myAdminServer.example.com:7001` using the username/password credentials `weblogic/password1`:

   ```
   connect("weblogic","password1","t3://myAdminServer.example.com:7001")
   ```

6.3.2 Viewing MSAS Instance Configuration Using WLST

You can display all of the configuration properties, with their values and groups, as specified in the current configuration document in the repository. If a property is not defined in the configuration document, then the default value defined for the product is displayed.

To view the configuration for an MSAS instance:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Use the `displayMSASConfiguration()` command to display the MSAS instance configuration.

   ```
   displayMSASConfiguration([instanceName=None])
   ```

   For example, to display the configuration for the instance named `myinstance1`, enter the following command:

   ```
   wls:/new_domain/serverConfig> displayMSASConfiguration('myinstance1')
   .
   .
   .
   Name: "client.clock.skew" Category: "Agent" Source: "default"
   Value: 0

   Name: "compliance.check" Category: "Agent" Source: "default"
   Value: true

   Name: "clock.skew" Category: "Agent" Source: "default"
   Value: 360000

   Name: "expire.time" Category: "Agent" Source: "default"
   Value: 300000

   Name: "cache.refresh.repeat" Category: "BeanAccessor" Source: "default"
   Value: 600000
   .
   ```

```
.
.
```

Note that the output displayed above shows the source of the configuration property settings as `default`, indicating the setting is from the default configuration document for the product. If you have modified a setting, it is saved in a configuration document for your instance, and will be reflected in the `SOURCE` field. For example:

```
Name: "max.connection.pool.per.host" Category: "ClientConfiguration" Source:
"myinstance1"
Value: 100
```

> **Note:** Some of the properties that are displayed in the output of the `displayMSASConfiguration` command are not supported in this release of Mobile Security Access Server and are reserved for future use. This applies specifically to the properties in the following categories:
>
> - ConfigManager
> - Identity
> - IssuedToken
> - ClassPathAccessor

6.3.3 Using the setMSASConfiguration WLST Command to Configure an MSAS Instance

Many of the instance-level configuration properties can be set using the `setMSASConfiguration` command, as described in this section. You do not need to use the `setMSASConfiguration` command in the context of a session.

To set or modify the configuration properties for the MSAS instance using the `setMSASConfiguration` command:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Optionally, use the `displayWSMConfiguration()` command to view the current configuration for the instance as described in "Viewing MSAS Instance Configuration Using WLST" on page 6-28.
3. Use the `setMSASConfiguration()` command to set the desired configuration properties.

   ```
   setMSASConfiguration
   (instanceName,categoryName,propertyName,[groupName=None],[PropertyValues=None])
   ```

   In this command:

   - `instanceName`—The MSAS instance for which the configuration is to be modified.
   - `categoryName`—The category to which the property belongs. The category is verified against the default set of properties to ensure it is valid. This field is required.
   - `propertyName`—The name of the property. The name is verified against the default set of properties to ensure it is valid. This field is required.

- `groupName`—The group containing the set of values to add in the configuration document. If the group exists, and this value is set to `None`, the group is removed. This field is optional.
- `propertyValues`—The array of values to set for a property or group in the configuration document. The default is None, which refers to an empty array list. This field is optional.

This command behaves as follows:

- If a property is already defined, you can update it by entering the updated value in the `propertyValues` field. The property is set to the new value.
- To clear a property setting, do not enter a value in the `propertyValues` field.

For example, to modify the maximum connection pool properties for outbound message settings:

```
wls:/base_domain/serverConfig>
setMSASConfiguration('myinstance1','ClientConfiguration','max.connection.pool.t
otal',None,['1000'])

A new property "max.connection.pool.total" within category
"ClientConfiguration" has been added.
The values "[1000]" have been added to property "max.connection.pool.total"
within category "ClientConfiguration".
Configuration properties associated with the MSAS instance "myinstance1" has
been updated.

wls:/base_domain/serverConfig>
setMSASConfiguration('myinstance1','ClientConfiguration','max.connection.pool.p
er.host',None,['100'])

A new property "max.connection.pool.per.host" within category
"ClientConfiguration" has been added.
The values "[100]" have been added to property "max.connection.pool.per.host"
within category "ClientConfiguration".
Configuration properties associated with the MSAS instance "myinstance1" has
been updated.
```

For example, to clear the maximum connections per host property:

```
wls:/base_domain/serverConfig>
setMSASConfiguration('myinstance1','ClientConfiguration','max.connection.pool.p
er.host',None,None)

Value elements removed from the property "max.connection.pool.total" within
category "ClientConfiguration".
The property "max.connection.pool.total" within category "ClientConfiguration"
has been removed.
Configuration properties associated with the MSAS instance "myinstance1" has
been updated.
```

Note: If you do not modify a configuration property, or you remove a property as shown in the example above, the default value is used at run time.

Refer to the subsequent sections for the list of configuration properties and the default values for each type of configuration.

6.3.4 Configuring an Identity Store Profile Using WLST

Mobile Security Access Server includes WLST commands that provide the ability to add an identity store profile to the MSAS instance, edit an existing profile, delete a profile, and set the default profile.

An identity store profile is a logical representation of a user repository. All user and group entities are present in this identity store. There can be multiple profiles associated with an MSAS instance, and one profile can be marked as the default against which all authentication and user profile queries will occur.

The identity store configuration is stored in an Identity Profile Document in the MSAS Repository.

When using WLST to create, modify, and delete identity store profile documents, you must execute the commands in the context of a session. Each session applies to a single identity store profile document only.

For more information about these commands, see "MSAS Identity Store Profile Commands" in *WebLogic Scripting Tool Command Reference for Identity and Access Management*.

The following sections describe how to use WLST in interactive mode for identity profile management:

- Creating an Identity Store Profile Using WLST
- Updating an Identity Profile Using WLST
- Deleting an Identity Profile

6.3.4.1 Creating an Identity Store Profile Using WLST

To create an identity store profile using WLST:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Start a repository session using the `beginRepositorySession` command.

   The `beginRepositorySession` command is used to create a session in which the repository will be modified. All creation, modification, or deletion commands must be performed in the context of a session. A session can only act on a single document.

   For example:

   ```
   wls:/base_domain/serverConfig> beginRepositorySession()

   Repository Session begun.
   ```

3. List the identity profile documents configured in an instance using the `displayIdentityProfile` command.

   ```
   displayIdentityProfile(instanceName, [profileName=None])
   ```

   In this command:

   - `instanceName`—The name of the MSAS instance for which you want to display the identity profiles. This field is required.
   - `profileName`—Name of a specific identity profile to be displayed. This field is optional. If no value is provided for the `profileName` argument, then all the identity profiles associated with the instance are displayed.

   For example:

```
wls:/base_domain/serverConfig> displayIdentityProfile('myinstance1')

No identity profiles exist for instance "myinstance1"
```

4. Use the `createIdentityProfile` command to create an identity store profile document.

   ```
   createIdentityProfile(instanceName, profileName, [description=None])
   ```

   In this command:

   - `instanceName`—The name of the MSAS instance for which you want to create an identity profile. This field is required.
   - `profileName`—Name for the identity profile.
   - `description`—Description of the identity profile. This argument is optional.

   For example:

   ```
   wls:/base_domain/serverConfig>
   createIdentityProfile('myinstance1','identity-profile1','Identity Profile 1')
   Identity profile "identity-profile1" for instance "myinstance1" is successfully
   created in session.
   Commit session will be required to reflect the operation in the repository.
   ```

   Notes: You cannot create multiple identity profiles within a session. You must commit the session, and then start a new session to create another identity profile.

5. To set the identity profile directory, use the `setIdentityProfileDirectory` command.

   ```
   setIdentityProfileDirectory(directoryType, hostport, bindDN, bindPass, baseDN,
   isSecure)
   ```

   In this command:

   - `directoryType`—Type of directory to use for the identity profile. Supported values are:
     - `OID` (Oracle Internet Directory)
     - `OUD` (Oracle Unified Directory)
     - `ACTIVE_DIRECTORY`
     - `ODSEE` (Oracle Directory Server Enterprise Edition)
     - `WLS_LDAP` (Embedded LDAP in WebLogic Server)
   - `hostport`—Host name and port of the server running the selected directory.
   - `bindDN`—DistinguishedName (DN) of the user to connect to the directory.
   - `bindPass`—Password for the Bind DN to connect to the directory.
   - `baseDN`—LDAP Searchbase under which all users and groups are located in the LDAP directory. For example, `cn=us, dn=mycompany, dc=com`.
   - `isSecure`—Flag (boolean) to indicate if the connection to the directory should be made over SSL. When set to `true`, the connection is configured over SSL.

For example:

```
wls:/base_domain/serverConfig>
setIdentityProfileDirectory('OID',['host1.mycompany.com:5678'],'cn=host,dn=myco
mpany,dn=com','welcome','cn=us,dn=mycompany,dn=com',false)

Directory information for identity profile set successfully.
```

6. To set the user information for an identity profile, use the `setIdentityProfileUser` command.

   ```
   setIdentityProfileUser(baseDN, loginIDAttribute, objectClassNames)
   ```

   In this command:

   – `baseDN`—Base DNs used to create or search for users. For example, `cn=users, dn=mycompany, dc=com`.

   – `loginIDAttribute`—Login ID for the user. Typically, this is the `uid` or `mail` attribute in the LDAP. In Active Directory, this refers to the `UserPrincipalName`.

   – `objectClassNames`—Fully qualified names of the schema classes used to represent users. Typically, this field is set to the standard LDAP objectclass `inetorgperson`.

   For example:

   ```
   wls:/base_domain/serverConfig>
   setIdentityProfileUser('cn=users,dc=mycompany,dc=com','uid',['inetorgperson'])

   User information for identity profile set successfully.
   ```

7. To set the group information for an identity profile, use the `setIdentityProfileGroup` command.

   ```
   setIdentityProfileGroup(baseDN, groupNameAttribute, objectClassNames)
   ```

   In this command:

   – `baseDN`—Base DNs used to create groups or enterprise roles. For example, `cn=group, dn=mycompany, dc=com`.

   – `groupNameAttribute`—Attribute that uniquely identifies the name of the enterprise group or role.

   – `objectClassNames`—List of objectclasses used to identify the enterprise roles or groups. Typically, this field is set to the standard LDAP standard objectclass `groupofuniquenames`. In Active Directory, this is `group`.

   For example:

   ```
   wls:/base_domain/serverConfig>
   setIdentityProfileGroup('cn=group,dc=mycompany,dc=com','cn',['groupofuniquename
   s'])

   Group information for identity profile set successfully.
   ```

8. Write the current contents of the session to the MSAS Repository using the `commitRepositorySession` command.

   For example:

   ```
   wls:/base_domain/serverConfig> commitRepositorySession()
   ```

```
create operation performed successfully on identity profile "identity-profile1"
for instance "myinstance1".
```

Alternatively, you can choose to cancel any changes by using the `abortRepositorySession` command, which discards any changes that were made to the repository during the session.

9. To set an identity profile as the default, use the `setMSASConfiguration` command.

   > **Note:** This step must be performed before you can use this identity profile at run time. For details about using this command, see "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

   For example:

   ```
   wls:/base_domain/serverConfig>
   setMSASConfiguration('myinstance1','IdentityProfile','name',None,['identity-pro
   file1'])

   The values "[identity-profile1]" have been added to property "name" within
   category "IdentityProfile".
   Configuration properties associated with the MSAS instance "myinstance1" has
   been updated.
   ```

10. To use this new identity store profile, you must restart the MSAS server.

6.3.4.2 Updating an Identity Profile Using WLST

To update an identity store profile using WLST:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Start a repository session using the `beginRepositorySession` command.

   The `beginRepositorySession` command is used to create a session in which the repository will be modified. All creation, modification, or deletion commands must be performed in the context of a session. A session can only act on a single document.

   For example:

   ```
   wls:/base_domain/serverConfig> beginRepositorySession()

   Repository Session begun.
   ```

3. List the identity profile documents configured in an instance using the `displayIdentityProfile` command.

   ```
   displayIdentityProfile(instanceName, [profileName=None])
   ```

   In this command:

   - `instanceName`—The name of the MSAS instance for which you want to display the identity profiles. This field is required.
   - `profileName`—Name of a specific identity profile to be displayed. This field is optional. If no value is provided for the `profileName` argument, then all the identity profiles associated with the instance are displayed.

   For example:

```
wls:/base_domain/serverConfig> displayIdentityProfile('myinstance1')

Identity Profiles for instance "myinstance1":
identity-profile1
```

4. To display the contents of the identity profile, use the `displayIdentityProfile` command with the `profileName` argument.

   For example:

```
wls:/base_domain/serverConfig>
displayIdentityProfile('myinstance1','identity-profile1')

Name : identity-profile1
Instance name : myinstance1
Description : Identity Profile 1

Directory Information:
        Directory Type : OID
        Bind DN : cn=host,dn=mycompany,dn=com
        Base DN : cn=us,dn=mycompany,dn=com
        Hosts : host1.mycompany.com:5335
        SSL Enabled :  false


User Information:
        Base DN : cn=users,dc=mycompany,dc=com
        Login ID Attribute : uid
        Object class names : inetorgperson


Group Information:
        Base DN : cn=group,dc=mycompany,dc=com
        Group Name Attribute : cn
        Object class names : groupofuniquenames


Commit session will be required to reflect the operation in the repository.
```

5. Select the identity profile to be updated using the `selectIdentityProfile` command.

```
selectIdentityProfile(instanceName, profileName)
```

   In this command:

   – `instanceName`—The name of the MSAS instance for which you want to edit the identity profile. This field is required.

   – `profileName`—Name for the identity profile to be edited.

   For example:

```
wls:/base_domain/serverConfig>
selectIdentityProfile('myinstance1','identity-profile’)


Identity profile "identity-profile1" for instance "myinstance1" selected for
modification.
```

6. To edit the profile directory, for example, use the `setIdentityProfileDirectory` command.

   ```
   setIdentityProfileDirectory(directoryType, hostport, bindDN, bindPass, baseDN,
   isSecure)
   ```

   In this command:

   – `directoryType`—Type of directory to use for the identity profile. Supported values are:

     – `OID` (Oracle Internet Directory)

     – `OUD` (Oracle Unified Directory)

     – `ACTIVE_DIRECTORY`

     – `ODSEE` (Oracle Directory Server Enterprise Edition)

     – `WLS_LDAP` (Embedded LDAP in WebLogic Server)

   – `hostport`—Host name and port of the server running the selected directory.

   – `bindDN`—DistinguishedName (DN) of the user to connect to the directory.

   – `bindPass`—Password for the Bind DN to connect to the directory.

   – `baseDN`—LDAP Searchbase under which all users and groups are located in the LDAP directory. For example, `cn=us, dn=mycompany, dc=com`.

   – `isSecure`—Flag (boolean) to indicate if the connection to the directory should be made over SSL. When set to `true`, the connection is configured over SSL.

   For example, to change the directory type from OID to WLS_LDAP:

   ```
   wls:/base_domain/serverConfig> setIdentityProfileDirectory('WLS_
   LDAP',['host1.mycompany.com:5678'],'cn=host,dn=mycompany,dn=com','welcome','cn=
   us,dn=mycompany,dn=com',false)

   Directory information for identity profile set successfully.
   ```

7. Optionally, display the contents of the identity profile again, using the `displayIdentityProfile` command with the `profileName` argument.

   > **Note:** When you execute this command within a session, the session changes are displayed. If you execute it outside of a session, the contents of the repository are displayed.

   For example:

   ```
   wls:/base_domain/serverConfig>
   displayIdentityProfile('myinstance1','identity-profile1')
   Identity profile "identity-profile1" for instance "myinstance1" is selected for
   update operation in session.

   Name : identity-profile1
   Instance name : myinstance1
   Description : Identity Profile 1


   Directory Information:
           Directory Type : WLS_LDAP
           Bind DN : cn=host,dn=mycompany,dn=com
   ```

```
        Base DN : cn=us,dn=mycompany,dn=com
        Hosts : host1.mycompany.com:5678
        SSL Enabled :  false

User Information:
        Base DN : cn=users,dc=mycompany,dc=com
        Login ID Attribute : uid
        Object class names : inetorgperson


Group Information:
        Base DN : cn=group,dc=mycompany,dc=com
        Group Name Attribute : cn
        Object class names : groupofuniquenames
```

8. Write the current contents of the session to the MSAS Repository using the `commitRepositorySession` command.

   For example:

   ```
   wls:/base_domain/serverConfig> commitRepositorySession()

   update operation performed successfully on identity profile "identity-profile1"
   for instance "myinstance1".
   ```

9. Optionally, use the `setMSASConfiguration` command to set the updated identity profile as the default. This step is only required if you want to use the updated identity profile as the default and it was not previously set as the default profile.

   For example:

   ```
   wls:/base_domain/serverConfig>
   setMSASConfiguration('myinstance1','IdentityProfile','name',None,['identity-pro
   file1'])

   The values "[identity-profile1]" have been added to property "name" within
   category "IdentityProfile".
   Configuration properties associated with the MSAS instance "myinstance1" has
   been updated.
   ```

10. Restart the MSAS server.

6.3.4.3 Deleting an Identity Profile

To delete an identity profile using WLST:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.

2. Start a repository session using the `beginRepositorySession` command.

   The `beginRepositorySession` command is used to create a session in which the repository will be modified. All creation, modification, or deletion commands must be performed in the context of a session. A session can only act on a single document.

   For example:

   ```
   wls:/base_domain/serverConfig> beginRepositorySession()

   Repository Session begun.
   ```

3. Optionally, list the identity profile documents configured in an instance using the `displayIdentityProfile` command.

   ```
   displayIdentityProfile(instanceName, [profileName=None])
   ```

   In this command:

   - `instanceName`—The name of the MSAS instance for which you want to display the identity profiles. This field is required.
   - `profileName`—Name of a specific identity profile to be displayed. This field is optional. If no value is provided for the `profileName` argument, then all the identity profiles associated with the instance are displayed.

   For example:

   ```
   wls:/base_domain/serverConfig> displayIdentityProfile('myinstance1')

   Identity Profiles for instance "myinstance1":
   identity-profile1
   ```

4. Delete the identity profile using the `deleteIdentityProfile` command.

   ```
   deleteIdentityProfile(instanceName, profileName)
   ```

   In this command:

   – `instanceName`—The name of the MSAS instance associated with the identity profile. This field is required.

   – `profileName`—Name of the identity profile to be deleted.

   For example:

   ```
   wls:/base_domain/serverConfig>
   deleteIdentityProfile('myinstance1','identity-profile1’)

   Identity profile "identity-profile1" for instance "myinstance1" is successfully
   deleted in session.
   Commit session will be required to reflect the operation in the repository.
   ```

5. Write the current contents of the session to the MSAS Repository using the `commitRepositorySession` command.

   For example:

   ```
   wls:/base_domain/serverConfig> commitRepositorySession()

   delete operation performed successfully on identity profile "identity-profile1"
   for instance "myinstance1".
   ```

6. If the identity profile that you deleted was set as the default profile for the instance, you need to run the `setMSASConfiguration` command to either clear the property if a different profile is not available, or set a different profile as the default profile.

   For example, to clear the property, enter:

   ```
   wls:/base_domain/serverConfig>
   setMSASConfiguration('myinstance1','IdentityProfile','name',None, None)

   Value elements removed from the property "name" within category
   "IdentityProfile".
   The property "name" within category "IdentityProfile" has been removed.
   Configuration properties associated with the MSAS instance "myinstance1" has
   ```

```
been updated.
```

To set a different profile as the default, enter:

```
wls:/base_domain/serverConfig>
setMSASConfiguration('myinstance1','IdentityProfile','name',None,['identity-pro
file2'])

The values "[identity-profile2]" have been added to property "name" within
category "IdentityProfile".
Configuration properties associated with the MSAS instance "myinstance1" has
been updated.
```

7. Restart the MSAS server.

6.3.5 Defining Trusted Issuers and Managing DN Lists Using WLST

SAML and JWT trusted issuers and DN lists are stored in trust configuration documents in the MSAS repository. To configure trusted issuers and DN lists, you must create a new document or edit an existing document in the repository.

When using WLST to create, modify, and delete token issuer trust documents, you must execute the commands in the context of a session. Each session applies to a single trust document only.

For more information about trusted issuers and DN lists, see "Configuring Trusted Issuers and DN Lists for Signing Certificates" on page 6-7.

The following sections describe how to use WLST commands in interactive mode to define trusted issuers and DN lists, how to import and export trust metadata, and how to revoke trust from a trusted issuer.

- Configuring Trusted Issuers and DN Lists Using WLST
- Deleting a Trusted Issuer Using WLST
- Deleting a Token Issuer Trust Document Using WLST
- Exporting and Importing Trust Configuration Using WLST
- Revoking Trust From Trusted Issuers Using WLST

6.3.5.1 Configuring Trusted Issuers and DN Lists Using WLST

To configure SAML and JWT trusted issuers and DN lists using WLST:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Start a repository session using the `beginRepositorySession` command.

   The `beginRepositorySession` command is used to create a session in which the repository will be modified. All creation, modification, or deletion commands must be performed in the context of a session. A session can only act on a single document.

   For example:

   ```
   wls:/base_domain/serverConfig> beginRepositorySession()

   Repository session begun.
   ```

3. List the token issuer trust documents in the repository using the `listWSMTokenIssuerTrustDocuments` command.

   ```
   listWSMTokenIssuerTrustDocuments(name=None, detail='false')
   ```

When used without any arguments, all the token issuer trust documents in the repository are listed. If you set the `detail` argument to `true`, the display name and the status of the document are also displayed. You can use the wildcard character * anywhere in the `name` argument. If no wildcard is specified, then the exact value entered for the `name` argument is used as the document name.

For example:

```
wls:/base_domain/serverConfig> listWSMTokenIssuerTrustDocuments('true')

Starting Operation listWSMTokenIssuerTrustDocuments ...

There are no token issuer trust documents in the repository.
```

4. Do one of the following:

   - If a token issuer trust domain document does not exist for your domain, create one using the `createWSMTokenIssuerTrustDocument` command. The `name` argument is required.

     Note: Once you execute this command, you must run the `setMSASConfiguration` command using the properties specified in the output of the `createWSMTokenIssuerTrustDocument` command.

     ```
     createWSMTokenIssuerTrustDocument(name, displayName)
     ```

     For example:

     ```
     wls:/base_domain/serverConfig> createWSMTokenIssuerTrustDocument('trust_
     t1')

     New Token Issuer Trust document named "trust_t1" created.
     To use the new document in the domain configuration, you must run the
     setConfiguration command where category = "TokenIssuerTrust", property name
     = "name" and value = "trust_t1".

     wls:/base_domain/serverConfig> setMSASConfiguration
     (’myinstance1’,'TokenIssuerTrust','name',None,['trust_t1'])

     The values "[trust_t1]" have been added to property "name" within category
     "TokenIssuerTrust".
     Configuration properties associated with the MSAS instance "myinstance1"
     has been updated.
     ```

   - If the token issuer trust document already exists for your domain, select the document for modification using the `selectWSMTokenIssuerTrustDocument` command. The `name` argument is required.

     ```
     selectWSMTokenIssuerTrustDocument(name)
     ```

     For example:

     ```
     wls:/base_domain/serverConfig> selectWSMTokenIssuerTrustDocument('trust_
     t1')

     Token Issuer Trust document named "trust_t1" selected in the session.
     ```

5. Optionally, specify a display name for the document using the `setWSMTokenIssuerTrustDisplayName` command. The display name is optional

but can be useful for describing the documents. Note that if you specified a display name for the document when you created it, you can use this command to change the display name if desired.

```
setWSMTokenIssuerTrustDisplayName(displayName)
```

For example:

```
wls:/base_domain/serverConfig> setWSMTokenIssuerTrustDisplayName('myinstance1
Trust Document')

Starting Operation setWSMTokenIssuerTrustDisplayName ...

Display Name of the document changed from null to myinstance1 Trust Document.
```

6. Add the trusted issuers and define trusted keys or a trusted DN list using the `setWSMTokenIssuerTrust` command.

```
setWSMTokenIssuerTrust(type, issuer, [trustedKeyIds=None])
```

In this command:

- `type`—The types of the tokens issued by the issuer and how the issuer signing certificates are identified with `trustedKeyIds`. Supported type values are shown in the following table.

Use this type value...	For this token type...	With this key type...	And this key identifier type
`dns.sv`	SAML SV	X509 certificate	DN
`dns.hok`	SAML HOK or Bearer	X509 certificate	DN
`dns.jwt`	JWT	X509 certificate	DN

- `issuer`—Name of the trusted issuer, such as `www.oracle.com`.
- `trustedKeyIds`—Optional argument used to specify the trusted key identifiers or the DN list for the issuer.

This command behaves as follows:

- If the trusted issuer already exists for the type specified, and you provide a list of DNs or aliases for the `trustedKeyIds` argument, the previous list is replaced with the new list. If you enter an empty set (`[]`) for the `trustedKeyIds` argument, then the list of DN values are deleted for the issuer.
- If the trusted issuer does not exist for the type specified and you specify a value for the `trustedKeyIds` argument, the issuer is created with the associated DN list. If you do not set the `trustedKeyIds` argument, a new issuer is created with an empty DN list.

In the following example, `www.yourcompany.com` is set as a trusted issuer. A DN list is not specified:

```
wls:/base_domain/serverConfig>
setWSMTokenIssuerTrust("dns.jwt","www.yourcompany.com",[])

Starting Operation setWSMTokenIssuerTrust ...

JWT trusted issuers successfully set
```

In the following example, `CN=weblogic, OU=Orakey, O=Oracle, C=US'` and `CN=orcladmin, OU=Doc, O=Oracle, C=US'` are set as DNs in the `dns.jwt` DN list for the `www.yourcompany.com` trusted JWT issuer:

```
wls:/base_domain/serverConfig>
setWSMTokenIssuerTrust('dns.jwt','www.yourcompany.com',
['CN=weblogic, OU=Orakey, O=Oracle’,
'CN=orcladmin, OU=Doc, O=Oracle, C=US'])

Starting Operation setWSMTokenIssuerTrust ...

JWT trusted issuers successfully set
```

7. Display the trusted issuer and DN list using the `displayWSMTokenIssuerTrust` command.

   ```
   displayWSMTokenIssuerTrust(type, issuer=None)
   ```

   When you specify a value for the `type` and `issuer` arguments, the DN lists for the issuer are displayed. If you do not specify an issuer name, all of the trusted issuers for the given type are listed.

   For example, to view the DN lists for the `www.yourcompany.com` trusted issuer:

   ```
   wls:/base_domain/serverConfig> displayWSMTokenIssuerTrust('dns.jwt',
   'www.yourcompany.com')

   Starting Operation displayWSMTokenIssuerTrust ...

   CN=weblogic, OU=Orakey, O=Oracle
   CN=orcladmin, OU=Doc, O=Oracle, C=US
   ```

   To view all of the trusted issuers for the type `dns.jwt`:

   ```
   wls:/base_domain/serverConfig> displayWSMTokenIssuerTrust('dns.jwt')

   Starting Operation displayWSMTokenIssuerTrust ...

   www.yourcompany.com
   www.oracle.com
   ```

8. Write the current contents of this session to the OWSM Repository using the `commitRepositorySession` command.

   For example:

   ```
   wls:/base_domain/serverConfig> commitRepositorySession()

   The tokenissuertrust trust_t1 is valid.
   Creating tokenissuertrust trust_t1 in repository.

   Repository session committed successfully.
   ```

Alternatively, you can choose to cancel any changes by using the `abortRepositorySession` command, which discards any changes that were made to the repository during the session.

For more information about these commands, see "Token Issuer Trust Configuration Commands" in *WebLogic Scripting Tool Command Reference for Identity and Access Management*.

6.3.5.2 Deleting a Trusted Issuer Using WLST

You can delete a trusted issue from a token issuer trust document in the repository using the `deleteWSMTokenIssuerTrust` command. The issuer must exist in the token issuer trust document selected in the session.

To delete a trusted issuer:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Start a repository session using the `beginRepositorySession` command.

   For example:

   ```
   wls:/base_domain/serverConfig> beginRepositorySession()

   Repository session begun.
   ```

3. List the token issuer trust documents in the repository using the `listWSMTokenIssuerTrustDocuments` command.

   ```
   listWSMTokenIssuerTrustDocuments(name=None, detail='false')
   ```

   When used without any arguments, all the token issuer trust documents in the repository are listed. If you set the `detail` argument to `true`, the display name and the status of the document are also displayed.

   For example:

   ```
   wls:/base_domain/serverConfig> listWSMTokenIssuerTrustDocuments(detail='true')

   Name         : oracle-default
   Display Name : Token Issuer Trust Properties
   Status       : DOCUMENT_STATUS_COMMITED

   Name         : trust_t1
   Display Name : myinstance1 Trust Document
   Status       : DOCUMENT_STATUS_COMMITED
   ```

4. Select the document for modification that contains the trusted issuer to be deleted using the `selectWSMTokenIssuerTrustDocument` command. The `name` argument is required.

   ```
   selectWSMTokenIssuerTrustDocument(name)
   ```

   For example:

   ```
   wls:/base_domain/serverConfig> selectWSMTokenIssuerTrustDocument('trust_t1')

   Token Issuer Trust document named "trust_t1" selected in the session.
   ```

5. Optionally, display the trusted issuers defined in the trust document using the `displayWSMTokenIssuerTrust` command.

   ```
   displayWSMTokenIssuerTrust(type, issuer=None)
   ```

   For example, to display all the trusted issuers for the JWT token type:

   ```
   wls:/base_domain/serverConfig> displayWSMTokenIssuerTrust('dns.jwt')

   Starting Operation displayWSMTokenIssuerTrust ...

   www.yourcompany.com
   www.oracle.com
   ```

6. Delete the desired token issuer, and its associated DN list if applicable, using the `deleteWSMTokenIssuerTrust` command.

   ```
   deleteWSMTokenIssuerTrust(type, issuer)
   ```

   For example, to delete the `www.yourcompany.com` trusted issuer for the JWT token type:

   ```
   wls:/base_domain/serverConfig>
   deleteWSMTokenIssuerTrust('dns.jwt','www.yourcompany.com')

   Starting Operation deleteWSMTokenIssuerTrust ...

   JWT trusted issuers deleted successfully.
   ```

7. Write the contents of the current session to the repository using the `commitRepositorySession` command.

   ```
   wls:/base_domain/serverConfig> commitRepositorySession()

   Repository session committed successfully.
   ```

   Alternatively, you can choose to cancel any changes by using the `abortRepositorySession` command, which discards any changes that were made to the repository during the session.

6.3.5.3 Deleting a Token Issuer Trust Document Using WLST

To delete a token issuer trust document from the repository:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Start a repository session using the `beginRepositorySession` command.

   For example:

   ```
   wls:/base_domain/serverConfig> beginRepositorySession()

   Repository session begun.
   ```

3. List the token issuer trust documents in the repository using the `listWSMTokenIssuerTrustDocuments` command.

   ```
   listWSMTokenIssuerTrustDocuments(name=None, detail='false')
   ```

   When used without any arguments, all the token issuer trust documents in the repository are listed. If you set the `detail` argument to `true`, the display name and the status of the document are also displayed.

   For example:

   ```
   wls:/base_domain/serverConfig> listWSMTokenIssuerTrustDocuments(detail='true')

   Name         : oracle-default
   Display Name : Token Issuer Trust Properties
   Status       : DOCUMENT_STATUS_COMMITED

   Name         : trust_t1
   Display Name : myinstance1 Trust Document
   Status       : DOCUMENT_STATUS_COMMITED
   ```

4. Delete the desired token issuer trust document using the `deleteWSMTokenIssuerTrustDocument` command. The `name` argument is required.

   ```
   deleteWSMTokenIssuerTrustDocument(name)
   ```

   For example:

   ```
   wls:/base_domain/serverConfig> deleteWSMTokenIssuerTrustDocument('trust_t1')
   Token Issuer Trust document named "trust_t1" deleted from the repository.
   ```

5. Write the contents of the current session to the repository using the `commitRepositorySession` command.

   ```
   wls:/base_domain/serverConfig> commitRepositorySession()

   Deleting tokenissuertrust trust_t1 from repository.

   Repository session committed successfully.
   ```

   Alternatively, you can choose to cancel any changes by using the `abortRepositorySession` command, which discards any changes that were made to the repository during the session.

6.3.5.4 Exporting and Importing Trust Configuration Using WLST

You can export and import the trust configuration (Issuers, DNs, token attribute rules) between systems using the `exportWSMTokenIssuerTrustMetadata` and `importWSMTokenIssuerTrustMetadata` commands. These commands do not need to be run in a session.

When you export trust configuration, it is exported to an XML file at a location that you specify. You can export all trusted issuers, or exclude specific issuers. You can then copy the file to another system, and then import the trust metadata from the XML file.

Exporting Trust Configuration

To export trust metadata, use the `exportWSMTokenIssuerTrustMetadata` command.

```
exportWSMTokenIssuerTrustMetadata(trustFile,excludeIssuers=None)
```

In this command:

- `trustFile`—Location of the XML file where the exported metadata will be stored.
- `excludeIssuers`—Optional argument used to specify the list of issuers for which the trust metadata should *not* be exported.

For example, to export the trust metadata and exclude www.oracle.com, use the following command:

```
wls:/base_domain/serverConfig>
exportWSMTokenIssuerTrustMetadata('/tmp/trustData.xml',['www.oracle.com'])

Starting Operation exportWSMTokenIssuerTrustMetadata ...

Configuration for trusted issuers successfully exported.
```

Importing Trust Configuration

To import the trust metadata for all trusted issuers, use the `importWSMTokenIssuerTrustMetadata` command.

```
importWSMTokenIssuerTrustMetadata(trustFile)
```

In this command:

- `trustFile`—Location of the XML file from which the metadata will be imported.

For example, to import the trust metadata from the file `/tmp/trustData.xml`, use the following command:

```
wls:/base_domain/serverConfig>
importWSMTokenIssuerTrustMetadata('/tmp/trustData.xml')

Starting Operation importWSMTokenIssuerTrustMetadata ...

Configuration for trusted issuers successfully imported.
```

6.3.5.5 Revoking Trust From Trusted Issuers Using WLST

You can revoke trust by removing trusted issuers and associated configurations (DNs and token attribute rules) using the `revokeWSMTokenIssuerTrust` WLST command. You can remove all trusted issuers, or specify specific issuers to be excluded from the list to be removed using the `excludeIssuers` argument. If no argument is passed, then all trusted issuers and their associated configuration are removed.

```
revokeWSMTokenIssuerTrust(excludeIssuers=None)
```

For example, to revoke the trust from all issuers except `www.oracle.com`, use the following command:

```
wls:/base_domain/serverConfig> revokeWSMTokenIssuerTrust(['www.oracle.com'])

Starting Operation revokeWSMTokenIssuerTrust ...

Configuration for trusted issuers successfully removed.
```

6.3.6 Configuring Message Security Using WLST

You can configure the MSAS keystore and message security settings such as clock skew and message expiration time using WLST commands. This configuration is described in the following sections:

- Managing the MSAS Keystore Using Keystore Service Commands
- Configuring the Signature and Encryption Keys in the MSAS Keystore Using WLST
- Configuring Security Settings Using WLST

6.3.6.1 Managing the MSAS Keystore Using Keystore Service Commands

The Keystore Service uses a dedicated set of command-line commands for keystore operations such as creating and managing keystores, exporting certificates, and generating keypairs. While their usage is similar, these commands are distinct from other WLST commands. Details about these commands and their usage is provided in "About Keystore Service Commands" in *Securing Applications with Oracle Platform Security Services*.

Before using the Keystore Service commands, you must execute the `getOpssService` command to obtain an OPSS service command object that enables you to execute the commands and get help.

The following procedure describes how to execute the keystore commands to manage the MSAS keystore.

1. Connect to the running server as described in "Accessing the MSAS WLST Commands" on page 6-27.

2. Execute the `getOpssService` command to access the Keystore Service commands:

   ```
   wls:/base_domain/serverConfig>svc = getOpssService(name='KeyStoreService')
   ```

3. Optionally, execute the help command `svc.help` to get a list of the available commands or help on a specific command. To get a list of the available commands, use `svc.help()`. To get help on a specific command, provide the command name as the argument. For example, to get help on the `createKeyStore` command, enter the following:

   ```
   wls:/base_domain/serverConfig> svc.help('createKeyStore')
   Description:
   Creates a new keystore.

   Syntax:
   svc.createKeyStore(appStripe='<stripe>', name='<keystore>',
   password='<password>',permission=true|false)
   svc=the service command object obtained through a call to getOpssService()
   appStripe= the name of the stripe in which keystore is created.
   name= the name of the keystore.
   password= Password of the keystore.
   permission= true if keystore is protected by permission only, false if
   protected by both permission and password.

   Example:
   svc.createKeyStore(appStripe='system', name='keystore1', password='<password>',
   permission=true)
   ```

4. Create a new keystore using the `createKeystore` command:

   > **Note:** The MSAS keystore is created automatically when you create an MSAS instance using the `configMSAS` script, or when you register a logical instance in the MSAS console. You only need to execute this command if you want to create the keystore before you create or register the instance.

   ```
   wls:/base_domain/serverConfig>
   svc.createKeyStore(appStripe='mynewinstance',name='keystore',password='',permis
   sion=true)
   Already in Domain Runtime Tree

   Keystore created
   ```

5. To generate a keypair, use the `generateKeyPair` command.

   ```
   svc.generateKeyPair(appStripe='<stripe>', name='<keystore>',
   password='<password>', dn='<distinguishedname>', keysize='<keysize>',
   alias='<alias>', keypassword='<keypassword>')
   ```

   For example:

   ```
   wls:/base_domain/serverConfig>
   svc.generateKeyPair(appStripe='mynewinstance',name='keystore',password='',dn='c
   n=orakey',keysize='1024',alias='orakey',keypassword='')
   Already in Domain Runtime Tree
   ```

```
Key pair generated
```

6. Optionally, list the keystores configured in the environment for all instances using the `listKeyStores` command:

```
wls:/base_domain/serverConfig> svc.listKeyStores(appStripe='*')
Already in Domain Runtime Tree

system/trust
system/castore
myinstance1/keystore
mynewinstance/keystore
```

7. To import an existing JKS keystore into the KSS keystore, use the `importKeyStore` command:

```
svc.importKeyStore(appStripe='<stripe>', name='<keystore>',
password='<password>', aliases='<comma-separated-aliases>',
keypasswords='<comma-separated-keypasswords>', type='<keystore-type>',
permission=true|false, filepath='<absolute_file_path>')
```

   For example:

```
wls:/base_domain/serverConfig>
svc.importKeyStore(appStripe='mynewinstance',name='keystore',password='',aliase
s='orakey',keypasswords='orakey',type='JKS',
permission=true,filepath='/path/default-keystore.jks')
Already in Domain Runtime Tree

Keystore imported. Check the logs if any entry was skipped.
```

6.3.6.2 Configuring the Signature and Encryption Keys in the MSAS Keystore Using WLST

When you create an MSAS instance, it creates the MSAS signature keystore by default, using the MSAS instance name as the stripe name, such as `kss://myinstance/keystore`. It also imports the necessary certificates and keys. Because this KSS keystore is defined at the instance level, the signature and encryption keys apply to all applications in the instance.

You can configure the encryption and signature keys for the MSAS keystore using the `setMSASConfiguration` command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

Table 6–2 lists the keystore properties that you can set to encryption and signature keys.

Table 6–2 MSAS Keystore Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
KeystoreConfig	keystore.enc.csf.key	N/A	Alias of the key used to encrypt and decrypt messages.
KeystoreConfig	keystore.sig.csf.key	N/A	Alias of the key used for storing the signature key in the keystore.
KeystoreConfig	keystore.pass.csf.key	N/A	Reserved for future use.

Table 6–2 (Cont.) MSAS Keystore Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
KeystoreConfig	keystore.csf.map	N/A	The map in the CSF used to locate the MSAS Keystore specific credentials. This property should not be changed.
KeystoreConfig	keystore.type	N/A	This value is configured when the MSAS instance is registered with the Mobile Security Manager and should not be changed.
KeystoreConfig	location	N/A	Location of the KSS keystore: kss://*msas_instance_name*/keystore. This value is configured when you create the MSAS instance and should not be changed.

6.3.6.3 Configuring Security Settings Using WLST

To configure the security settings, use the `setMSASConfiguration` command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

Table 6–3 lists the MSAS instance-level configuration properties that you can set for security policy enforcement.

Table 6–3 Security Policy Enforcement Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
Agent	client.clock.skew	0	Tolerance of time, in seconds, that is used to calculate the `NotBefore` and `NotOnOrAfter` conditions for SAML and JWT token generation. Together, these conditions define the lower and upper boundaries to limit the validity of the token.
Agent	clock.skew	360000	Tolerance of time differences between client and server machines. For example, when timestamps are sent across in a message to a service that follows a different time zone, this property allows for the specified time tolerance. For more information about this property, see the description for the **Clock Skew** property in "Configuring Security Settings" on page 6-12.
Agent	expire.time	300000	Duration of time before a message expires after its creation. This property is used in cases where a timestamp is sent across in the token to verify if the timestamp has expired or not. For more information about this property, see the description for the **Message Expiration Time** property in "Configuring Security Settings" on page 6-12.
Agent	compliance.check	true	Reserved for future use.
Agent	nonce.ttl	28800000	Reserved for future use.
Agent	allow.all.xpaths	false	Reserved for future use.
Agent	use.unified.fault.code	true	Reserved for future use.

6.3.7 Configuring the Cache Refresh Time Using WLST

To configure cache management, use the `setMSASConfiguration` command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

Table 6–4 lists the MSAS-instance level configuration properties that you can set to configure the policy cache.

Table 6–4 Cache Management Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
BeanAccessor	cache.refresh.repeat	86400000	The number of milliseconds to wait between cache refreshes.
BeanAccessor	cache.refresh.initial	600000	Reserved for future use.
BeanAccessor	failure.retry.count	2	Reserved for future use.
BeanAccessor	cache.refresh.batch.size	10	Reserved for future use.
BeanAccessor	failure.retry.delay	5000	Reserved for future use.
BeanAccessor	missing.retry.delay	15000	Reserved for future use.
BeanAccessor	usage.record.delay	30000	Reserved for future use.

6.3.8 Configuring the Authentication Endpoints Using WLST

You can configure the authentication endpoints used for initial authentication using the `setMSASConfiguration` command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

The following sections describe the configuration properties for the different authentication endpoints.

- Configuring the KINIT and PKINIT Authentication Endpoint Using WLST
- Configuring the OAuth2 Confidential Client Endpoint Using WLST
- Configuring the OAuth2 Mobile Client Endpoint Using WLST
- Configuring the Crypto Service Endpoint Using WLST

6.3.8.1 Configuring the KINIT and PKINIT Authentication Endpoint Using WLST

Table 6–5 lists the configuration properties that you can set to configure KINIT and PKINIT authentication for the MSAS instance.

Table 6–5 KINIT/PKINIT Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
Krb5Configuration	default_realm	N/A	Name of the realm to be used as the default realm.
Krb5Configuration	default_tkt_enctypes	N/A	Session key encryption type that the client should use when making an initial authentication request (AS-REQ). For a list of the supported types, refer to the MIT Kerberos Documentation, "Encryption types" at `http://web.mit.edu/Kerberos/krb5-1.12/doc/admin/conf_files/kdc_conf.html#encryption-types`.
Krb5Configuration	default_tgs_enctypes	N/A	Session key encryption type that the client should use when requesting a service ticket from the TGS (TGS_REQ). For a list of the supported types, refer to the MIT Kerberos Documentation, "Encryption types" at `http://web.mit.edu/Kerberos/krb5-1.12/doc/admin/conf_files/kdc_conf.html#encryption-types`.

Table 6–5 (Cont.) KINIT/PKINIT Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
Krb5Configuration	pkinit_anchors	N/A	Trust anchors that the KDC will use when evaluating the trust of the client certificate. `pkinit_anchors` points to the keystore alias which should be the first certificate in the certificate chain. This is a mandatory property when PKINIT is used.
Krb5Configuration	pkinit_anchors_file	N/A	Reserved for internal use.
Krb5Configuration	logging.krb5	N/A	Log location for the Kerberos configuration messages. Valid options are STDERR or a log file name and path.
Krb5Configuration	logging.kcm	N/A	Log location for the KCM configuration messages. Valid options are STDERR or a log file name and path.
Krb5Configuration	realms.kdc	N/A	Name for the Kerberos realm. The realm name must match the REALM name defined in the Active Directory setup.
Krb5Configuration	realms.default_domain	N/A	Default domain for the realm. Typically the domain name is in lower case, for example `example.com`.
Krb5Configuration	domain	N/A	DNS domain name.

6.3.8.2 Configuring the OAuth2 Confidential Client Endpoint Using WLST

Table 6–6 lists the configuration properties that you can set to configure OAuth2 Confidential Client authentication for the MSAS instance.

Table 6–6 OAuth2 Confidential Client Configuration Property for an MSAS Instance

Category	Property Name	Default	Description
OAuth2ConfidentialClientConfiguration	service.profile.endpoint	N/A	OAuth Service Profile Endpoint to which the MSAS server creates JWT User Token and OAM Tokens for OAuth2 Confidential Client Authentication flow. For example: `http://host:port/ms_oauth/oauth2/endpoints/oauthservice`

6.3.8.3 Configuring the OAuth2 Mobile Client Endpoint Using WLST

Table 6–7 lists the configuration properties that you can set to configure OAuth2 Mobile Client authentication for the MSAS instance.

Table 6–7 OAuth2 Mobile Client Configuration Property for an MSAS Instance

Category	Property Name	Default	Description
OAuth2MobileClientConfiguration	service.profile.endpoint	N/A	OAuth Service Profile Endpoint to which the MSAS server creates JWT User Token and OAM Tokens for OAuth2 Mobile Client Authentication flow. For example: `http://host:port/ms_oauth/oauth2/endpoints/oauthservice`

6.3.8.4 Configuring the Crypto Service Endpoint Using WLST

Table 6–7 lists the configuration properties that you can set to configure the key rollover feature in the PKI Crypto Service for the MSAS instance.

Table 6–8 Crypto Service Configuration Property for an MSAS Instance

Category	Property Name	Default	Description
CryptoServiceConfiguration	archive.key.aliases	N/A	Alias for the keystore containing archived private keys to be used by the key rollover feature. If left unset, the key rollover support feature is not enabled. If you enter an alias in this field, key rollover is enabled, and the alias specified is added to the list of aliases used by the rollover feature. A decryption error will not be returned by the crypto service unless decryption of a given ciphertext fails with all archived private keys.

6.3.9 Configuring System Settings Using WLST

You can configure system settings such as outbound message settings, proxy servers, load balancing, and SSL using WLST `setMSASConfiguration` command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

The following sections describe the configuration properties that you can set for the different system settings.

- Configuring Outbound Message Settings Using WLST
- Configuring Proxy Server Settings Using WLST
- Configuring Server Settings Using WLST
- Configuring SSL Settings Using WLST

6.3.9.1 Configuring Outbound Message Settings Using WLST

Table 6–9 lists the configuration properties that you can set to configure outbound message settings for an MSAS instance.

Table 6–9 Outbound Message Setting Properties for an MSAS Instance

Category	Property Name	Default	Description
ClientConfiguration	max.connection.pool.per.host	25	Maximum number of connections in a pool, per host, that a client can handle.
ClientConfiguration	max.connection.pool.total	512	Maximum number of connections in a pool that a client can handle.
ClientConfiguration	idle.connection.pool.timeout	180000	Maximum time in milliseconds a client will keep idle connections in the pool.
ClientConfiguration	connection.timeout	20000	Maximum time in milliseconds a client can wait when connecting to a back-end host.
ClientConfiguration	request.timeout	60000	Maximum time in milliseconds a client can wait for a response.

Table 6–9 (Cont.) Outbound Message Setting Properties for an MSAS Instance

Category	Property Name	Default	Description
ClientConfiguration	request.longtimeout	1200000	Maximum time in milliseconds that a client can wait for a response.
ClientConfiguration	max.request.retry	5	Number of times a request will be retried before an error occurs because of a network exception.
ClientConfiguration	ssl.security.level	loose	SSL security level for outbound calls to back-end resources. For more information, see "Configuring SSL Between MSAS and Back-End Resources" on page 7-5.

6.3.9.2 Configuring Proxy Server Settings Using WLST

Table 6–10 lists the configuration properties that you can set to configure the proxy server used for outbound calls to the internet through MSAS for back-end applications and services.

Table 6–10 Proxy Server Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
ProxyServer	name	N/A	Name of proxy server to uniquely identify it.
ProxyServer	host	N/A	Host name of proxy server.
ProxyServer	port	N/A	Port number of proxy server.
ProxyServer	csf.key	N/A	Name of the CSF key that has credentials for authenticating to the proxy server. This property is optional.
ProxyServer	non.proxy.hosts	localhost 127.0.0.1	List of hosts that will not use the proxy server. It supports the asterisk * wildcard, but only as a suffix and prefix.

6.3.9.3 Configuring Server Settings Using WLST

Table 6–11 lists the configuration properties such as load balancing URLs and service principal name mapping that you can set for an MSAS instance to configure the server.

Table 6–11 Server Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
ServerSettings	spn.mapping	N/A	Maps a URL with Service Principal Name. Service Principal Name is required for NTLM and SPNEGO. URL supports wildcard using "*" and can go anywhere in the URL, for example, `http*://example.host*80/*` or `*.example.org`. SPN should be in form of `<SPN_SERVICECLASS>/<SPN_HOSTNAME>."`
ServerSettings	lbr.url	N/A	Non-SSL URL for a front ending load balancer, for example `http://lbr.example.org:80`.
ServerSettings	lbr.ssl.url	N/A	SSL URL for a front-ending load balancer, for example `https://lbr.example.org:443`.

6.3.9.4 Configuring SSL Settings Using WLST

Table 6–12 lists the configuration properties that you can set to specify the location of the SSL keystore and truststore for an MSAS instance.

Table 6–12 SSL Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
ServerSettings	ssl.truststore.location	N/A	Location of SSL trust store. Only KSS keystore type is supported so the value must be a KSS URI.
ServerSettings	ssl.keystore.location	N/A	Location of the SSL keystore used for inbound SSL connections to MSAS and as the MSAS identity keystore. Only KSS keystore type is supported so the value must be a KSS URI.

6.3.9.5 Configuring Access Log Settings Using WLST

Table 6–12 lists the configuration properties that you can set to enable or disable access logs for the MSAS instance.

Table 6–13 SSL Configuration Properties for an MSAS Instance

Category	Property Name	Default	Description
ServerSettings	access.log.enabled	`true`	Enables or disables the access logs for the MSAS server. By default, this property is enabled. For more information about access logs, see "Configuring MSAS Access Logs" on page 9-4.
ServerSettings	access.log.format	N/A	Reserved for future use.

6.3.10 Configuring the SToken Expiry Time

By default, the session token (SToken) expiry time is set to 34800000ms (approximately 9.6 hours). You can adjust the SToken expiry time using the `setMSASConfiguration` WLST command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29, using `MSASConfig` as the `CategoryName` and `stoken.expiry.time` as the `propertyName`.

For example, to change the SToken expiry time to 5 minutes, use the following command:

```
setMSASConfiguration('myinstance1','MSASConfig', 'stoken.expiry.time', None,
['300000'])
```

Note the following:

- The expiry time and skew time give an actual expiration for an SToken.
- The SToken expiry time that you configure cannot exceed the maximum expiry time set by any token except the KINIT/PKINIT SToken.
- The limit for the KINIT/PKINIT SToken maximum expiry time can be set as the default expiry time.

6.3.11 Configuring the MSAS Heartbeat Using WLST

The heartbeat is a process used to check if an MSAS instance is running normally and is reacting to the heartbeat polling requests (reliable UDP protocol). It is a standalone process that runs on each machine that is hosting one or more MSAS instances. By default, if an MSAS instance is dead or is not responding to heartbeat requests for any reason, the heartbeat process will try to restart that target instance.

The heartbeat process runs completely in the background. When the first MSAS instance is started on the host machine, the heartbeat process is started silently and registers the MSAS instance so that it can send polling messages. When additional instances are started on that machine, they are registered into the same heartbeat process. The polling and restart behavior is targeted per instance on each machine. When an individual instance is stopped, (`stopServer.sh`), the heartbeat process will unregister that instance from its polling group. If all instances on a machine are stopped, the heartbeat process quits.

The heartbeat process is configuration driven. All of the heartbeat configuration properties are managed by the Mobile Security Manager (MSM) at the logical instance level. You configure the MSAS heartbeat using the `setMSASConfiguration` WLST command as described in "Using the setMSASConfiguration WLST Command to Configure an MSAS Instance" on page 6-29.

The Mobile Security Manager should be installed and running before updating the heartbeat configuration. Table 6–14 lists the configuration properties that you can set to configure the MSAS heartbeat.

Table 6–14 Heartbeat Configuration Properties for an MSAS instance

Category	Property Name	Default	Description
MSASConfig	heartbeat.enabled	true	Flag used to enable or disable the heartbeat function.
MSASConfig	heartbeat.restartEnabled	true	Controls whether to restart the MSAS instance if the number of failures exceeds the threshold (value set for `heartbeat.maxRetry`).
MSASConfig	heartbeat.port	7777	Local listening port for registering MSAS instances and MSAS responses.
MSASConfig	heartbeat.frequency	5	The frequency, in seconds, to send heartbeat requests.
MSASConfig	heartbeat.maxRetry	5	Maximum number of retries if there no response from the MSAS instance.
MSASConfig	heartbeat.logFileName	heartbeat.log	Log file name for the heartbeat process. The heartbeat log file is created in the following directory: `*instance_root*/*instance_name*/log`
MSASConfig	heartbeat.logLevel	20	Log level for the heartbeat. When set to 20 (coarse logging) only messages such as MSAS register/deregister or a target instance being restarted are logged. For finer grained logging for debugging purposes, set this value to 10. In this case, every heartbeat message is logged.

6.3.12 Configuring Additional Server Settings Using WLST

Table 6–15 lists the configuration properties for additional server settings that you can set using the `setMSASConfiguration` command.

Table 6–15 Configuration Properties for Additional Server Settings

Category	Property Name	Default	Description
MSASConfig	msm.csf-key	N/A	Reserved for internal use.
MSASConfig	msm.url	N/A	Reserved for internal use.
MSASConfig	msas.id	N/A	Reserved for internal use.
MSASConfig	sync.interval	60000	Maximum time in milliseconds for the server to check for a synchronization event. The synchronization event is generated when the **Synchronize** button corresponding to the MSAS instance in clicked in the MSAS console. For more information about using the **Synchronize** button, see "Synchronizing MSAS Instance Configuration" on page 2-2.

Table 6–15 (Cont.) Configuration Properties for Additional Server Settings

Category	Property Name	Default	Description
MSASConfig	server.workerThread.corePoolSize	8	Core thread pool size.
MSASConfig	server.workerThread.maxPoolSize	1024	Maximum thread pool size.
MSASConfig	server.keepAlive	true	HTTP connection keep alive.
MSASConfig	server.readBufferSize	40960	Read buffer size, in bytes, for HTTP connection.
MSASConfig	server.writeBufferSize	16384	Write buffer size, in bytes, for HTTP connection.
MSASConfig	server.connectionTimeout	16384	Timeout, in milliseconds, when connecting to the server. The request will wait for the specified time before it times out due to a connection error.
MSASConfig	server.socketTimeout	180000	Timeout, in milliseconds, in server socket wait time.
MSASConfig	server.writeTimeout	180000	Write timeout, in milliseconds, for the server to write back to clients. Increase it if there are slower clients but increasing it too much will have an effect on connections.
MSASConfig	server.connectionBacklog	4096	Number of connections allowed in the backlog.
MSASConfig	server.clientSocketTimeout	180000	Timeout, in milliseconds, in client socket wait time.
MSASConfig	server.http.maxRequestHeaderSize	32768	Max request header size, in bytes.
MSASConfig	security.clientAuthenticationRequired	NO	Client authentication required. Valid values: ■ YES—Client authentication required. ■ NO—Client authentication not required. ■ MAY—Client authentication optional.
MSASConfig	security.keystoreAlias	msasidentity	Keystore alias.

6.3.13 Configuring the Credential Store Using WLST

You can create, update, and delete credentials in the credential store using WLST commands.

To use WLST commands to configure the credential store:

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. To create a credential in the credential store, use the `createCred` command.

```
createCred(map="<mapname>", key="<keyname>", user="<userName>",
password="<password>",[desc="<description>"])
```

   In this command:

   – `map`—Specifies a map name or folder in the credential store. For MSAS, this is the MSAS instance name. This field is required.

- `key`—Specifies a key name.
- `user`—Specifies the credential user name.
- `password`—Specifies the credential password.
- `description`—Description of the credential. This argument is optional.

For example:

```
wls:/base_domain/serverConfig>
createCred('myinstance1','key','demoUser','demoPassword')
```

3. To update a credential in the credential store, use the `updateCred` command.

```
updateCred(map="<mapname>", key="<keyname>", user="<userName>",
password="<password>", desc="<description>")
```

In this command:

- `map`—Specifies a map name or folder in the credential store. For MSAS, this is the MSAS instance name. This field is required.
- `key`—Specifies a key name.
- `user`—Specifies the credential user name.
- `password`—Specifies the credential password.
- `description`—Description of the credential. This argument is optional.

For example:

```
wls:/base_domain/serverConfig>
updateCred('myinstance1','demoKey','demoUser','newPassword')
```

4. To delete a credential from the credential store, use the `deleteCred` command.

```
deleteCred(map="<mapname>", key="<keyname>")
```

In this command:

- `map`—Specifies a map or folder in the credential store. For MSAS, this is the MSAS instance name.
- `key`—Specifies a key name.

For example, to delete a credential with map name `myinstance1` and key name `demoKey`:

```
wls:/base_domain/serverConfig>  deleteCred('myinstance1','demoKey')
```

For more information about these commands, see the following topics:

- "Managing the Credential Store" in *Securing Applications with Oracle Platform Security Services*
- "Security Commands" in *WebLogic Scripting Tool Command Reference for Identity and Access Management*

6.4 Advanced Kerberos Configuration

You can use the MSAS console pages to configure the properties of the Kerberos `krb5.conf` file to enable KINIT and PKINIT authentication. You can add, edit, and delete Kerberos realms, add and delete DNS domains, specify Kerberos encryption types, specify the logging location for the Kerberos and KCM messages, and enable the use of PKINIT trust anchors. For details, see "Configuring KINIT and PKINIT Authentication" on page 6-14.

However, if you need more advanced configuration and customization, you must create the `krb5.conf` file manually. For example, you may need to customize the file to point to specific domain controllers or to accommodate environments with alternate UPN suffixes. Once you save this file, the changes are persisted and it takes precedence over the file created using the console.

> **Note:** You should only create the `krb5.conf` file as described in this section if advanced configuration is required. Updates to this file are not supported by the MSAS console, therefore any changes that you make using the console are ignored.
>
> Do not manually edit the `krb5.conf` file in the *`instance_name`*`/config` directory as that is intended for use only by the MSAS console.

The following sections describe how to create and edit the file in these situations.

6.4.1 Creating the Kerberos Configuration File Manually

To create the krb5.conf file:

1. Create a `default` directory under the following location:

   `instance_root``/`*`instance_name`*`/config/`

   where *`instance_root`* is the root directory you specified when you created the instance, and *`instance_name`* is the name of the instance. By default, *`instance_root`* is *`MW_HOME`*`/instances`, and *`MW_HOME`* is the Middleware home directory in which you installed Mobile Security Access Server.

   For example, `/home/instances/myinstance1/config/default`

2. Create a `krb5.conf` file using the settings required for your environment as described in the MIT Kerberos Documentation at `http://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/krb5_conf.html`.

   For example:

```
[libdefaults]
  default_realm = EXAMPLE.COM
  default_tkt_enctypes = arcfour-hmac-md5
  default_tgs_enctypes = arcfour-hmac-md5
  kdc_timeout = 30
  max_retries = 1

[appdefaults]
  pkinit_anchors = FILE:instance_root/instancename/config/pkinit_anchors.cer

[logging]
  krb5 = STDERR
  kcm = STDERR

[realms]
  EXAMPLE.COM = {
  kdc = example.com
  default_domain = example.com
```

```
    }

  [domain_realm]
    .example.com = EXAMPLE.COM
```

3. Save the `krb5.conf` file to the *`instance_root/instance_name`*`/config/default` directory that you created in step 1.

   Note that once you create and save the `krb5.conf` file to this directory, it takes precedence over the one created using the console and any subsequent updates using the console are ignored.

6.4.2 Adding Multiple Active Directory Domains

To add additional Active Directory forests and domains, edit the Kerberos configuration file that you created manually at *`instance_root/instance_name`*`/config/default/krb5.conf` and add the realms to the `realms` section and the domain-to-realm mapping to the `domain_realm` section using the following syntax.

```
[realms]
  <KRB_REALM_NAME> = {
  kdc = <domain_name>
  default_domain = <domain_name>
  }

[domain_realm]
 .<domain_name> = <KRB_REALM_NAME>
```

For example:

```
 [realms]
  EXAMPLE1.COM = {
  kdc = example1.com
  default_domain = example1.com
  }
  EXAMPLE2.COM = {
  kdc = example2.com
  default_domain = example2.com
  }
[domain_realm]
  .example1.com = EXAMPLE1.COM
  .example2.com = EXAMPLE2.COM
```

6.4.3 Targeting Specific Domain Controllers

By default, after installation Mobile Security Access Server is configured to find the domain controllers for a specific domain by doing a DNS look up. The entries in the `realms` section for each domain in the *`instance_root/instance_name`*`/config/default/krb5.conf` file will look something like the following:

```
EXAMPLE.COM={
  kdc=example.com
  default_domain=example.com
}
```

You can configure Mobile Security Access Server to point to specific domain controllers for a given domain. Use a separate `kdc` line for each domain controller. For example:

```
EXAMPLE.COM = {
  kdc = dc1.example.com
  kdc = dc2.example.com
  default_domain = example.com
}
```

By default when there are multiple domain controllers configured, Mobile Security Access Server will try each of them in order. You can configure Mobile Security Access Server to try the individual domain controllers in random order by adding the statement `random_fallback = true` to the realm configuration. For example:

```
EXAMPLE.COM = {
  kdc = dc1.example.com
  kdc = dc2.example.com
  random_fallback = true
  default_domain = example.com
}
```

6.4.4 Adding Alternate UPN Suffixes

An alternate User Principal Name (UPN) suffix occurs when the domain in the UPN after the @ symbol is different from the Windows domain where the user resides, or any other Windows domain that can refer authentication requests to the user's domain.

For environments using accounts with alternate UPN suffixes and Windows password (KINIT), it is necessary to configure Mobile Security Access Server to perform Kerberos authentication using what are known as Enterprise Accounts. To turn on support for alternate UPN suffixes:

1. Open and edit the `krb5.conf` file at in the following location:

   `instance_root`/*`instance_name`*`/config/default/krb5.conf`

2. Add the following configuration line at the end of the `libdefaults` section:

```
[libdefaults
....]
   enterprise=true
```

Note: When using this flag it is important to set the `default_realm` parameter in the `libdefaults` section to point to the root domain that is below all sub-domains that contain users that need to authenticate.

For example, for a Windows forest comprised of a root domain `example.com` with two sub-domains `sub1.domain.com` and `sub2.domain.com`, the `default_realm` parameter should be set to `example.com`."

6.5 Manually Configuring OAuth2 Client Authentication

For the Mobile Security Access Server to authenticate users against Oracle Access Manager and retrieve Oracle Access Manager and OAuth tokens for integrated single sign on, the Mobile Security Access Server must be registered as an OAuth Confidential Client with the Oracle Access Manager OAuth Service. This configuration is completed automatically when you configure an MSAS instance using the Mobile Security Access Server configuration script, `configMSAS`.

However, you can also perform this configuration using the Oracle Access Management console and WLST. You can also use the procedures described in these sections to verify that the configuration is completed properly.

The following sections describe how to do so.

- Configuring OAuth2 Confidential Client Authentication
- Configuring OAuth2 Mobile Client Authentication

6.5.1 Configuring OAuth2 Confidential Client Authentication

There are three primary steps to configuring OAuth2 Confidential Client authentication.

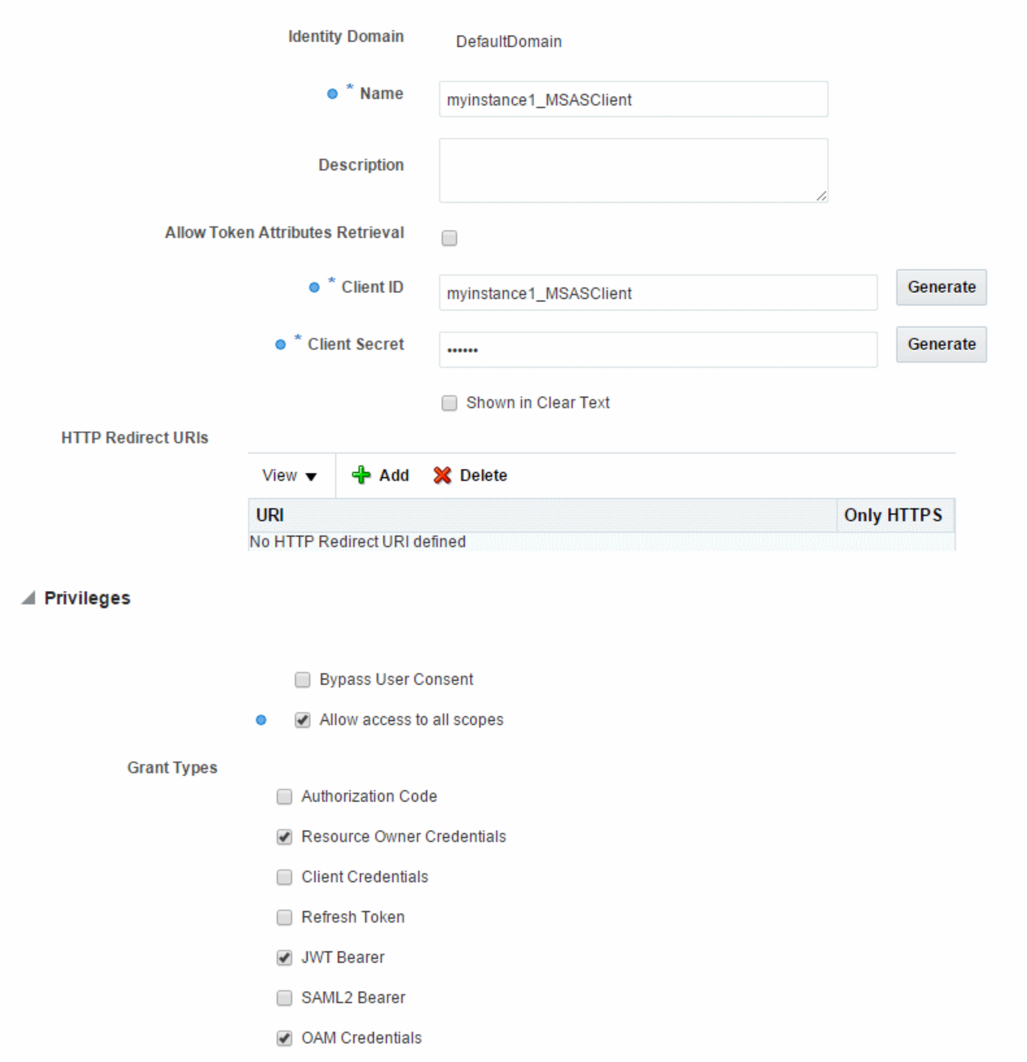
Step 1. Create the OAuth2 Confidential Client Profile

1. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
2. Click the Mobile OAuth Services icon. The Mobile OAuth Identity Domains page opens in a new tab.
3. Select the **Default Domain** in the table. The Mobile Identity Domain Configuration page opens in a new tab.
4. Click **Clients**, then in the OAuth Web Clients section, click the **Create** icon. The OAuth Web Client Configuration page opens in a new tab.
5. Enter a name, client ID, and client secret in the appropriate fields.

Field	Description
Name	Enter a name for the OAuth client using the format `*MSAS_Instance_ID_*MSASClient`, for example `myinstance1_MSASClient`.
Client ID	Enter a client ID using the format `*MSAS_Instance_ID_*MSASClient`, for example `myinstance1_MSASClient`.
Client Secret	Enter a password for the client. Make note of the password because you will need to provide it when configuring the credential store.

6. Expand the Privileges section, if it is not already expanded.
7. Select **Allow access to all scopes**, then in the Grant Types section, select **Resource Owner Credentials**, **JWT Bearer**, and **OAM Credentials**. The OAuth Web Client Configuration page is shown in Figure 6–5.

Figure 6–5 OAuth Web Client Configuration Page



Identity Domain: DefaultDomain

* Name: myinstance1_MSASClient

Description:

Allow Token Attributes Retrieval □

* Client ID: myinstance1_MSASClient

* Client Secret: ••••••

□ Shown in Clear Text

HTTP Redirect URIs

URI	Only HTTPS
No HTTP Redirect URI defined	

Privileges

□ Bypass User Consent

☑ Allow access to all scopes

Grant Types

□ Authorization Code

☑ Resource Owner Credentials

□ Client Credentials

□ Refresh Token

☑ JWT Bearer

□ SAML2 Bearer

☑ OAM Credentials

For detailed information about this page, see "Understanding the Mobile Clients Configuration Page" in *Administrator's Guide for Oracle Access Management*.

8. Click **Create** at the top of the page.

Step 2. Configure OAuth2 Confidential Client Service Profile Endpoint in the MSAS Instance

1. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
2. In the Mobile Security Access Server section, click **Environments**.

   The MSAS Environments page opens in a new tab.
3. Click **MSAS** or **Instances** in the MSAS tile.

   The MSAS Instances Summary page opens in a new tab.
4. Click the instance name or **Configure** in the tile for the desired instance.

   The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

5. Click the **Authentication Endpoints** tab.
6. In the OAuth2 Confidential Client section, enter the Service Profile Endpoints for the OAuth2 Confidential Client in the **Endpoint** field. For example: `http://`*`host:port`*`/ms_oauth/oauth2/endpoints/oauthservice`.
7. Click **Apply** at the top of the page.

Step 3. Add OAuth2 Confidential Client Credential to the CSF Using WLST

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Use the `createCred` WLST command to add the confidential client credential created in the previous steps to the credential store.

   ```
   createCred(map="<mapname>", key="<keyname>", user="<userName>",
   password="<password>",[desc="<description>"])
   ```

   For example, to add the credential for the instance named `myinstance1`, using the key `oauth2.confidential.client.credentials` with the client ID/password as `myinstance1_MSASClient/password1`, and the description as `OAuth2 Confidential Client Credential`:

   ```
   createCred('myinstance1','oauth2.confidential.client.credentials','myinstance1_
   MSASClient','password1','OAuth2 Confidential Client Credential')
   ```

   The username and password specified here must match the client ID and secret key that you provided when creating the OAuth web clients.

6.5.2 Configuring OAuth2 Mobile Client Authentication

There are three primary steps to configuring OAuth2 Mobile Client authentication.

Step 1. Create the OAuth2 Mobile Client Profile

1. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
2. Click the Mobile OAuth Services icon. The Mobile OAuth Identity Domains page opens in a new tab.
3. Select the **Default Domain** in the table. The Mobile Identity Domain Configuration page opens in a new tab.
4. Click **Clients**, then in the OAuth Mobile Clients section, click the **Create** icon. The OAuth Mobile Client Configuration page opens in a new tab.
5. In the **Name** and **Client ID** fields, enter a name and client ID using the format *`MSAS_Instance_ID`*`_OracleContainer`, for example `myinstance1_OracleContainer`.
6. Expand the Privileges section, if it is not already expanded.
7. Select **Allow access to all scopes**, then in the Grant Types section, select the following options:
   - **Resource Owner Credentials**
   - **Client Credentials**
   - **JWT Bearer**
   - **OAM Credentials**
   - **Client Verification Code**

For detailed information about this page, see "Understanding the Mobile Clients Configuration Page" in *Administrator's Guide for Oracle Access Management*.

8. Expand the Mobile Service Settings section and perform the following configuration:

   - Select **Override the default settings**.
   - In **Supported Platforms**, select **iOS** and **Android**.
   - For **iOS Security Level**, select **Standard**.
   - Clear the **Enable Server Side Single-Sign-On** checkbox.

9. Click **Create** at the top of the page.

Step 2. Configure OAuth2 Mobile Client Service Profile Endpoint in the MSAS Instance

1. From the Oracle Access Management home page, select the **Mobile Security** tab from the list of tabs at the top of the page.
2. In the Mobile Security Access Server section, click **Environments**.

   The MSAS Environments page opens in a new tab.

3. Click **MSAS** or **Instances** in the MSAS tile.

   The MSAS Instances Summary page opens in a new tab.

4. Click the instance name or **Configure** in the tile for the desired instance.

   The MSAS Instance Configuration page displays in a new tab. The tab name is the name of the instance.

5. Click the **Authentication Endpoints** tab.
6. In the OAuth2 Mobile Client section, enter the Service Profile Endpoints for the OAuth2 Mobile Client in the **Endpoint** field. For example: `http://`*`host:port`*`/ms_oauth/oauth2/endpoints/oauthservice`.
7. Click **Apply** at the top of the page.

Step 3. Add OAuth2 Mobile Client Credential to the CSF Using WLST

1. Start WLST as described in "Accessing the MSAS WLST Commands" on page 6-27.
2. Use the `createCred` WLST command to add the OAuth mobile client credential created in the previous steps to the credential store.

   For example, to add the mobile client credential for the instance named `myinstance1`, using the key `oauth2.mobile.client.id` with the client ID as `myinstance1_OracleContainer`, any value for the password since the password is not used for Mobile Client authentication, and the description as `OAuth2 Mobile Client Credential`:

   ```
   createCred('myinstance1','oauth2.mobile.client.id','myinstance1_
   OracleContainer','myinstance1_OracleContainer','OAuth2 Mobile Client
   Credential')
   ```

   The Client ID used in this command must match the Client ID that you used when you created the OAuth2 Mobile Client as described in Step 1, in this example `myinstance1_OracleContainer`. The password can be any value as it is not used in this configuration.

6.6 Configuring Single Sign-On (SSO) for OAM WebGate and Oracle WSM Protected Resources

If your environment includes resources protected by Oracle Web Services Manager (Oracle WSM) and/or WebGates for Oracle Access Manager (OAM), you can configure Mobile Security Access Server to provide single sign-on capability for those resources. The procedure is described in the following steps.

Step 1: Create an MSAS Instance

You can create and configure an MSAS instance using the `configMSAS` command as described in "Configuring an MSAS Instance" in *Installing Oracle Mobile Security Access Server*.

Step 2: Configure the OAuth2 Confidential and Mobile Client Endpoints

You can configure the OAuth2 endpoints using the MSAS console pages as described in the following sections:

- "Configuring OAuth2 Confidential Client Authentication" on page 6-18
- "Configuring Oracle Access Manager Mobile and Social (OAMMS) Authentication" on page 6-20

When configuring the authentication endpoint URL, you must enter the complete URL for the external OAuth server, for example:

```
http://example.com:14100/ms_oauth/oauth2/endpoints/oauthservice
```

Step 3: Create a forward proxy application for the Oracle Access Manager login page and secure it with access policies

To do so:

1. Create a forward proxy application as described in "Creating a Proxy Application" on page 3-5.
2. In the application, define a proxy URL for the Oracle Access Manager login page. In the Host URL field, enter the URL for the OAM login page, for example:

   ```
   http://host:port/oam/server
   ```

3. Select the URL, and attach the following policies to the policy enforcement endpoints as described in "Attaching Policies and Assertions to Proxy Applications" on page 5-4:

   - On-Request—`oracle/http_session_token_verify_policy`
   - Invoke-Proxy—`oracle/http_bmax_oam_client_policy`

   For details about these policies, see "oracle/http_session_token_verify_policy" and "oracle/http_bmax_oam_client_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

Step 4: Ensure that a forward proxy application exists for single-sign-on to OAM WebGate protected resources

When you create an MSAS instance, a reserved application named Default URL is created to protect all forward proxy requests by default. If you have not modified or deleted this application, your WebGate protected resources will be protected by the policies attached to the Default URL application. If you have modified or deleted this application, then you need to create a forward proxy application and attach specific policies to the policy enforcement endpoints as follows:

1. Create a forward proxy application as described in "Creating a Proxy Application" on page 3-5.
2. In the application, define a proxy URL for the WebGate.
3. Select the URL, and attach the following policies to the policy enforcement endpoints as described in "Attaching Policies and Assertions to Proxy Applications" on page 5-4:
   - On-Request—`oracle/http_session_token_verify_policy`
   - Invoke-Proxy—`oracle/multi_token_client_policy`

   For details about these policies, see "oracle/http_session_token_verify_policy" and "oracle/multi_token_client_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

Step 5: Ensure that a forward proxy application exists for single-sign-on to Oracle WSM protected resources

When you create an MSAS instance, a reserved application named Default URL is created to protect all forward proxy requests by default. If you have not modified or deleted this application, your Oracle WSM protected resources will be protected by the policies attached to the Default URL application. If you have modified or deleted this application, then you need to create a forward proxy application and attach specific policies to the policy enforcement endpoints as follows:

1. Create a forward proxy application as described in "Creating a Proxy Application" on page 3-5.
2. In the application, define a proxy URL for the Oracle WSM-protected resource.
3. Select the URL, and attach the following policies to the policy enforcement endpoints as described in "Attaching Policies and Assertions to Proxy Applications" on page 5-4:
   - On-Request—`oracle/http_session_token_verify_policy`
   - Invoke-Proxy—attach one of the following:
     - `oracle/multi_token_client_policy`—This should be the default policy if Oracle WSM has been configured to trust OAuth/JWT tokens issued by Oracle Access Manager.
     - `oracle/http_jwt_token_client_policy`—This policy can be used in advanced configurations where Oracle WSM has been configured to trust JWT tokens issued by MSAS.

   For details about these policies, see "oracle/http_session_token_verify_policy" and "oracle/multi_token_client_policy"in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server* and "oracle/http_jwt_token_client_policy" in *Security and Administrator's Guide for Web Services*.

6.7 Configuring Single Sign-On for Kerberos and NTLM Protected Resources

If your environment includes resources protected by Kerberos and/or NTLM, you can configure Mobile Security Access Server to provide single sign-on capability for those resources. The procedure is described in the following steps.

Step 1: Create an MSAS Instance

You can create and configure an MSAS instance using the `configMSAS` command as described in "Configuring an MSAS Instance" in *Installing Oracle Mobile Security Access Server*.

Step 2: Configure the Kerberos Authentication Endpoints

You can configure the Kerberos authentication (KINIT and/or PKINIT) endpoints using the MSAS console pages as described in the following sections:

- "Configuring KINIT and PKINIT Authentication" on page 6-14
- "Advanced Kerberos Configuration" on page 6-57

Step 3: Ensure that a forward proxy application exists for single-sign-on to Kerberos protected resources

When you create an MSAS instance, a reserved application named Default URL is created to protect all forward proxy requests by default. If you have not modified or deleted this application, your Kerberos protected resources will be protected by the policies attached to the Default URL application. If you have modified or deleted this application, then you need to create a forward proxy application and attach specific policies to the policy enforcement endpoints as follows:

1. Create a forward proxy application as described in "Creating a Proxy Application" on page 3-5.
2. In the application, define a proxy URL for the Kerberos protected resource.
3. Select the URL, and attach the following policies to the policy enforcement endpoints as described in "Attaching Policies and Assertions to Proxy Applications" on page 5-4:
   - On-Request—`oracle/http_session_token_verify_policy`
   - Invoke-Proxy—attach one of the following:
     - `oracle/multi_token_client_policy`
     - `oracle/http_bmax_spnego_client_policy`

For reference information about these policies, see "oracle/http_session_token_verify_policy", "oracle/multi_token_client_policy", and "oracle/http_bmax_spnego_client_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

Step 4: Ensure that a forward proxy application exists for single-sign-on to NTLM protected resources

When you create an MSAS instance, a reserved application named Default URL is created to protect all forward proxy requests by default. If you have not modified or deleted this application, your NTLM protected resources will be protected by the policies attached to the Default URL application. If you have modified or deleted this application, then you need to create a forward proxy application and attach specific policies to the policy enforcement endpoints as follows:

1. Create a forward proxy application as described in "Creating a Proxy Application" on page 3-5.
2. In the application, define a proxy URL for the NTLM protected resource.
3. Select the URL, and attach the following policies to the policy enforcement endpoints as described in "Attaching Policies and Assertions to Proxy Applications" on page 5-4:

- On-Request—`oracle/http_session_token_verify_policy`
- Invoke-Proxy—attach one of the following:
  - `oracle/multi_token_client_policy`
  - `oracle/http_ntlm_token_client_policy`

For reference information about these policies, see "oracle/http_session_token_verify_policy", "oracle/multi_token_client_policy", and "oracle/http_ntlm_token_client_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

6.8 Configuring an MSAS Instance as a WebGate

Mobile Security Access Server can be configured to serve as a WebGate in your environment. A WebGate is a web-server plug-in for Oracle Access Manager (OAM) that intercepts HTTP requests and protects resources by URL. When you create and configure an MSAS instance using the `configMSAS` tool as described in "Configuring an MSAS Instance" in *Installing Oracle Mobile Security Access Server*, a WebGate profile is automatically created using the responses that you provide to the OAM prompts. This WebGate profile specifies which resources will be protected by OAM. To configure the MSAS instance to act as a WebGate, you attach an OAM security policy to virtual and proxy applications in the instance.

The high-level flow in this scenario is as follows:

- The MSAS instance receives an HTTP request for access to a URL resource.
- If the resource is protected by the predefined OAM policy, MSAS routes the request to OAM for authentication.
- OAM presents its login page to the user for authentication.
- User provides credentials and OAM authenticates the user against the configured identity store.
- If the user is authenticated, OAM creates a session and sends an authentication response with a session token to MSAS.
- MSAS allows user access to the resource.

To configure MSAS as a WebGate, you must attach the predefined OAM security policy to the URL resources defined in virtual and proxy applications in the instance. To do so:

1. Create a virtual or proxy application in the instance as described in the following sections:
   - "Creating a Virtual Application" on page 3-3
   - "Creating a Proxy Application" on page 3-5
2. Open the URL Policy Configuration page for the resource to be protected as described in the following sections:
   - "Attaching Policies and Assertions to Virtual Applications" on page 5-2
   - "Attaching Policies and Assertions to Proxy Applications" on page 5-4
3. Attach the following policy to the On-Request endpoint for the URL:

   ```
   oracle/http_oam_authentication_service_policy
   ```

   For details about this policy, see "oracle/http_oam_authentication_service_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

4. Click **Validate**, then **Apply** to save the changes.

Note: If you need to update the WebGate profile in any way, you can do so using the OAM console as described in "Configuring and Managing Registered OAM Agents Using the Console" in *Administrator's Guide for Oracle Access Management*.

The MSAS instance will be listed, using the instance name, on the **WebGates** tab of the Search SSO Agents page. You may need to click **Search** to display the list of WebGates.

7

Configuring the SSL Keystore and Truststore

This chapter provides an overview of the SSL keystore and truststore, and describes how to configure SSL between MSAS and other system components. Topics include:

7.1 Understanding the SSL Keystore and Truststore

Mobile Security Access Server supports an SSL keystore and SSL truststore. The SSL keystore holds the identity key for the server and the SSL truststore serves as the repository for trusted certificates. The SSL truststore is used for trusting or authenticating client certificates (for two-way SSL).

Configuration of the SSL keystore and truststore is performed when you execute the `idmConfigTool`. For details about running the `idmConfigTool`, see "Configuring the Identity Store for the MSAS Instance" in *Installing Oracle Mobile Security Access Server*.

When you execute the `idmConfigTool`, it:

- Creates the SSL keystore and truststore for the MSAS instance.
- Creates an SSL key which is signed by a self-signed CA (certificate authority).
- Imports SSL certificates into the truststore that are required for Mobile Device Management/Mobile Application Management (MDM/MAM) flow and for two-way SSL communication with mobile clients.

> **Note:** Mobile Security Access Server supports only the OPSS KSS keystore for the SSL keystore and truststore.

7.1.1 SSL Keystore and Truststore Locations

The SSL keystore and truststores are created at the logical instance level in the following locations:

- SSL keystore—`kss://`*`msas_instance_id`*`/sslkeystore` (KSS stripe)

  This location is identified by the configuration property:

  - Category: `ServerSettings`
  - Property Name: `ssl.keystore.location`

- SSL truststore—`kss://`*`msas_instance_id`*`/ssltruststore` (KSS stripe).

  This location is identified by the configuration property:

  - Category: `ServerSettings`
  - Property Name: `ssl.truststore.location`

For details about setting these properties, see "Configuring Server Settings Using WLST" on page 6-53.

> **Note:** In both KSS stripes, *`msas_instance_id`* is the name of the logical MSAS instance with which the keystore and truststore are associated. Do not change these locations.

7.1.2 Managing the SSL Keystore and Truststore

You can manage the SSL keystore and truststore using the MSAS console pages in the OAM console. Using the console you can import keys into the SSL keystore, or generate keys. You can also import certificates into the SSL truststore. For details, see "Configuring the SSL Keystore and Truststore" on page 6-25.

For advanced management of the KSS SSL keystore and truststore, you can also use Keystore Service commands provided by Oracle Platform Security Service (OPSS). The Keystore Service uses a dedicated set of command-line commands for keystore operations such as creating and managing keystores, exporting certificates, and generating keypairs. While their usage is similar, these commands are distinct from other WLST commands. Details about these commands and their usage are provided in "About Keystore Service Commands" in *Securing Applications with Oracle Platform Security Services*.

> **Note:** By default, two-way SSL is not enabled on the MSAS server, and is controlled by the configuration property:
>
> - Category: `MSASConfig`
> - Property Name: `security.clientAuthenticationRequired`
>
> The settings for this property are as follows:
>
> - `NO` (the default)—Client authentication is not required and two-way SSL is disabled at the MSAS server level.
> - `YES`—Client authentication is required and two-way SSL is enabled at the MSAS server level. In this case, all clients must provide client certificates with each request, regardless of whether the URLs they are accessing require a client certificate or not.
> - `MAY`—Client authentication optional.
>
> For details about setting this property, see "Configuring Additional Server Settings Using WLST" on page 6-55.

No additional SSL configuration is required for communication between the following components:

- MSAS and the Mobile Security Manager
- MSAS and Oracle Access Manager and OAuth Server

For SSL configuration details between other components, refer to the following sections:

- Configuring SSL Between the Mobile Device and MSAS
- Configuring SSL Between MSAS and Back-End Resources
- Configuring SSL Between MSAS and the Identity Store

7.2 Configuring SSL Between the Mobile Device and MSAS

By default, SSL is mandatory in MSAS and the SSL port is always enabled for one-way SSL. You configure the SSL port when you run the MSAS configuration tool `configMSAS` to create the MSAS instance. The SSL keystore, which is configured when you run the `idmConfigTool`, will contain one key, signed by the self-signed certificate authority (CA), that is used as the identity key for the instance. For details about running the `configMSAS` tool and the `idmConfigTool`, refer to the following topics in *Installing Oracle Mobile Security Access Server*:

- "Configuring an MSAS Instance"
- "Configuring the Identity Store and Keystores for the MSAS Instance"

If the SSL keystore has more than one key, then the alias of the identity key must be specified in the configuration property `MSASConfig:security.keystoreAlias`. Keys can be imported or generated using the MSAS console or WLST commands.

For scenarios such as PKINIT-based authentication and mobile device registration, the client certificate is requested and authenticated, therefore the issuer certificate chains must be trusted in the SSL truststore. The client certificate must have a Subject Alternative Name extension that contains the User Principal Name (UPN) of the client.

You can import the signer's certificate chain for the client certificate into the SSL truststore using the MSAS console as described in "Configuring the SSL Keystore and Truststore" on page 6-25. You can also use OPSS Keystore Service commands to import the signer's certificate chain for the client certificate into the SSL truststore. Details about these commands and their usage are provided in "About Keystore Service Commands" in *Securing Applications with Oracle Platform Security Services*.

7.2.1 Obtaining a Trusted Certificate and Importing it into the SSL Keystore

The default identity key of the MSAS instance is signed by a self-signed CA. You can replace the certificate with one signed by a well known CA, if required, as described in the following procedure:

1. Export the existing KSS keystore entry into a JKS keystore using the Keystore Service commands:

   a. Connect to the running server as described in "Accessing the MSAS WLST Commands" on page 6-27.

   b. Execute the `getOpssService` command to access the Keystore Service commands:

   ```
   wls:/base_domain/serverConfig>svc = getOpssService(name='KeyStoreService')
   ```

   c. Export the KSS keystore using the `exportKeyStore` command:

   ```
   svc.exportKeyStore(appStripe='<msas-id>', name='sslkeystore',
   password='<keystore-password>', aliases='<msas-id>_msasidentity',
   keypasswords='<key-password>', type='JKS',filepath='/tmp/<msas-id>_
   sslkeystore.jks')
   ```

The password and keypasswords arguments in the command apply to the JKS keystore and key password. This password is used to protect the exported key and the JKS keystore.

2. Generate the server certificate request to create a Certificate Signing Request (CSR) file using the keytool -certreq command:

```
keytool -keystore /tmp/<msas-id>_sslkeystore.jks -storepass <keystore-password>
-alias <msas-id>_msasidentity -certreq -file /tmp/msasidentity.csr -keypass
<key-password>
```

3. Submit the CSR file to a CA. The CA will authenticate the request, issue a certificate for the MSAS instance, and return the certificate and a certificate chain.
4. Update the MSAS identity certificate on the server. To do so you must first update the JKS keystore that you exported in Step 1, and then import the JKS keystore into the KSS keystore.
   a. Import the new CA certificate chain into the JKS keystore. This is required to create a complete chain.

   ```
   keytool -keystore /tmp/<msas-id>_sslkeystore.jks -import -file <CA_
   CERT>.crt -alias ca -storepass <keystore-password>
   ```

   b. Import the updated MSAS identity certificate into same JKS keystore. This command assumes that the signed certificate is available at /tmp/msasidentity.crt.

   ```
   keytool -keystore /tmp/<msas-id>_sslkeystore.jks -import -file
   /tmp/msasidentity.crt -alias <msas-id>_msasidentity -storepass
   <keystore-password>
   ```

   c. Import the JKS keystore into the KSS keystore using the Keystore Service commands. Connect to the running server as described in "Accessing the MSAS WLST Commands" on page 6-27, then execute the following commands:

   ```
   svc = getOpssService(name='KeyStoreService')

   svc.deleteKeyStoreEntry(appStripe='<msas-id>',name='sslkeystore',password='
   ', alias='<msas-id>_msasidentity', keypassword='')

   svc.importKeyStore(appStripe='<msas-id>',name='sslkeystore',password='<keys
   tore-password>', aliases='<msas-id>_msasidentity',
   keypasswords='<key-password>', type='JKS',permission=true,
   filepath='/tmp/<msas-id<_keystore.jks')
   ```

5. Restart the MSAS server.

Note: Any changes in the SSL keystore or truststore require that you restart the MSAS server.

7.2.2 Downloading the MSAS Identity Certificate into the Mobile Device

The entity signing the MSAS identity certificate must be trusted in the mobile device. If not, the mobile device will not be able to connect to MSAS. If the entity signing the certificate is a well known certificate authority (CA), then ensure that the signer is already trusted in the mobile device.

To simplify the process of downloading the MSAS identity certificate chain from the SSL keystore, MSAS provides the following URL:

```
https://msas_host:msas_port/bmax/msas_cert[n].pem
```

where [*n*] represents a certificate in the chain from 0 to 4. The certificate at index zero is the issuer of the server certificate, and the certificate at index one is the issuer of the certificate at index zero and so on.

For example:

```
https://msas_host:msas_port/bmax/msas_cert0.pem
https://msas_host:msas_port/bmax/msas_cert1.pem
```

> **Note:** Because the SSL keystore contains the single self-signed root certificate, by default `http://msas_host:msas_port/bmax/msas_cert0.pem` provides the root certificate to install on the device.

7.3 Configuring SSL Between MSAS and Back-End Resources

By default, certificate authentication for back-end servers is turned off on MSAS. As a result, the SSL certificates of back-end resources are not required to be trusted in the MSAS SSL truststore and no configuration is required.

This behavior is controlled by the configuration property:

- Category: `ClientConfiguration`
- Property Name: `ssl.security.level`

The default value of this property is `loose`. Changing the value of this property to `strict` will require that the SSL certificates of the back-end resources be trusted in MSAS SSL truststore, otherwise MSAS will not be able to connect to them.

For information about setting this and other properties in the `ClientConfiguration` category, see "Configuring Outbound Message Settings Using WLST" on page 6-52.

> **Note:** Two-way SSL with back-end resources is not supported in this release.

7.4 Configuring SSL Between MSAS and the Identity Store

If the identity store is configured on an SSL port, then the SSL certificate for the identity store must be trusted in MSAS SSL truststore. You can import the self-signed SSL certificate or the signer’s certificate chain for the identity store (required for production environments) using the MSAS console, as described in "Configuring the SSL Keystore and Truststore" on page 6-25.

You can also use OPSS Keystore Service commands to import the self-signed certificate or signer’s certificate chain for the identity store into the SSL truststore. Details about these commands and their usage are provided in "About Keystore Service Commands" in *Securing Applications with Oracle Platform Security Services*.

> **Note:** Any changes in the SSL keystore or truststore require that you restart the MSAS server.

8 Managing Policies and Assertion Templates

This chapter includes the following sections:

- Overview of Policy and Assertion Template Management
- Managing Policies
- Validating Policies
- Managing Policy Assertion Templates
- Enabling or Disabling Policies and Assertions
- Defining Multiple Policy Alternatives (OR Groups)
- Recommended Naming Conventions for Policies
- Recommended Naming Conventions for Assertion Templates

8.1 Overview of Policy and Assertion Template Management

The following sections provide an overview of policy and assertion template management.

- Building Policies Using Policy Assertions
- Predefined Policies and Assertion Templates

8.1.1 Building Policies Using Policy Assertions

A policy is expressed as one or more **policy assertions** representing an application URLs capabilities or requirements. A policy assertion is the smallest unit of a policy that performs a specific action for the request and response operations. For example, a policy assertion may stipulate that a request to an application URL be encrypted. Likewise, a policy assertion can define the maximum message size that an application URL can accept.

8.1.2 Predefined Policies and Assertion Templates

There is a set of predefined policies and assertion templates that are automatically available. The predefined policies are based on common best practice policy patterns used in customer deployments.

You can immediately begin attaching these predefined policies to the URLs in the MSAS applications. You can edit and configure the predefined policies or create a new policy by making a copy of one of the predefined policies.

Predefined policies are constructed using assertions based on predefined assertion templates. You can create new assertion templates, as required.

For more information about the predefined policies and assertion templates, see:

- "Predefined Policies" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*
- "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*

8.2 Managing Policies

You manage policies from the Access Policies page. From this page you can:

- Search for specific policies or types of policies
- View policies
- Create, edit, and delete policies
- Import or export policies to or from the repository
- Add assertions or OR groups to a policy
- Version a policy

The following sections describe how to manage policies.

- Viewing Access Policies
- Searching for Policies
- Viewing the Details of a Policy
- Creating and Editing a Policy
- Exporting and Importing Policies
- Adding Assertions to a Policy
- Adding an OR Group to a Policy
- Versioning Policies
- Deleting a Policy

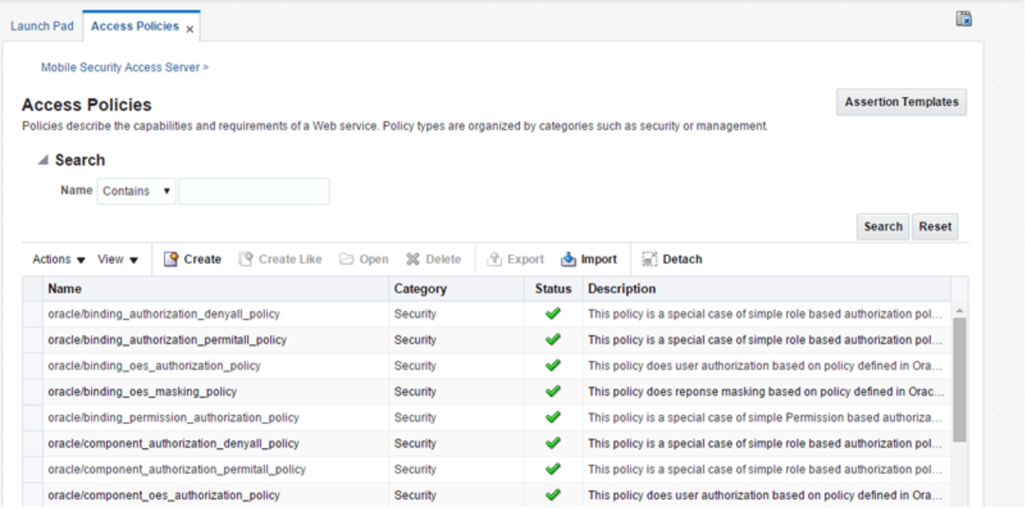
8.2.1 Viewing Access Policies

You view policies from the Access Policies page by performing the following steps:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

Figure 8–1 Access Policies Page



Access Policies

Policies describe the capabilities and requirements of a Web service. Policy types are organized by categories such as security or management.

Name	Category	Status	Description
oracle/binding_authorization_denyall_policy	Security	✓	This policy is a special case of simple role based authorization pol...
oracle/binding_authorization_permitall_policy	Security	✓	This policy is a special case of simple role based authorization pol...
oracle/binding_oes_authorization_policy	Security	✓	This policy does user authorization based on policy defined in Ora...
oracle/binding_oes_masking_policy	Security	✓	This policy does reponse masking based on policy defined in Orac...
oracle/binding_permission_authorization_policy	Security	✓	This policy is a special case of simple Permission based authoriza...
oracle/component_authorization_denyall_policy	Security	✓	This policy is a special case of simple role based authorization pol...
oracle/component_authorization_permitall_policy	Security	✓	This policy is a special case of simple role based authorization pol...
oracle/component_oes_authorization_policy	Security	✓	This policy does user authorization based on policy defined in Ora...

8.2.2 Searching for Policies

In the Access Policies page, you can reduce the number of policies that are returned by specifying the appropriate search criteria. To do so:

1. In the Search pane, specify the criteria to use in the search.

   In the **Name** field, enter a policy name or part of a policy name and select the operator to use to refine the search. Available operators are Starts with, Ends with, Equals, and Contains. For example, to search for message protection policies only, select the **Contains** operator, and enter `message` in the **Name** field.

   You can use percent `%` as a wildcard, any place in the name. Asterisk `*` is not recognized as a wildcard and is treated as plain text. Searches are case-insensitive.

2. Click **Search**.

   The Policies table is refreshed to include only those policies that match the specified search criteria.

8.2.3 Viewing the Details of a Policy

Use the following procedure to view the details of a policy.

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.

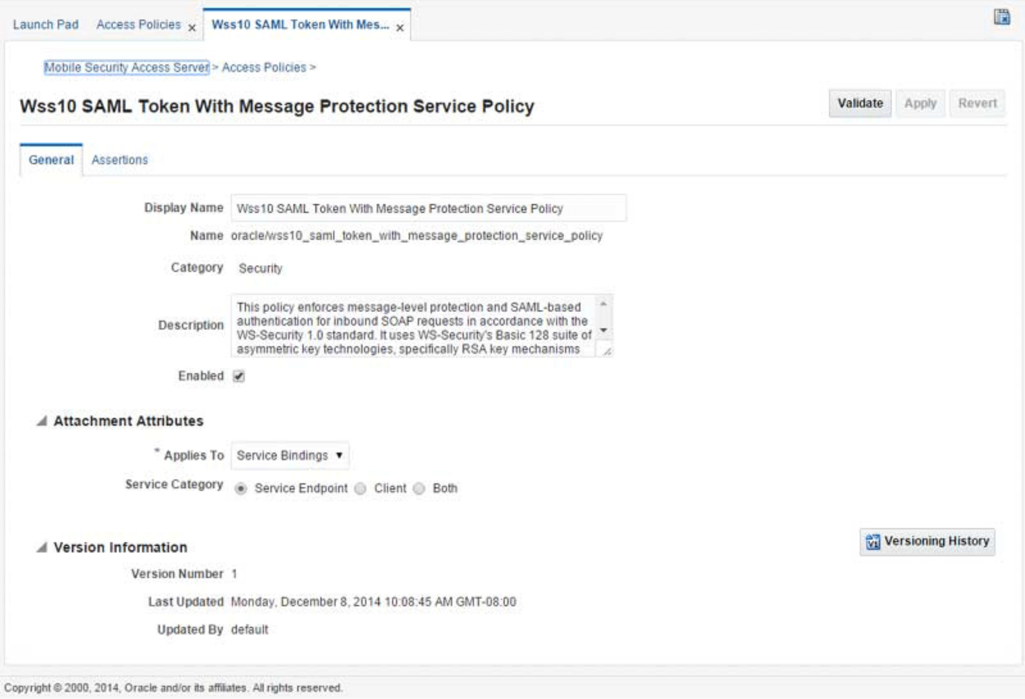
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab. Optionally, refine the list of policies displayed using Search, as described in "Searching for Policies" on page 8-3.

3. Select the policy to be viewed from the list of policies and click **Open**. Alternatively, select **Actions** and then **Open**.

   Figure 8–2 displays the Policy Details page for the `oracle/wss10_saml_token_with_message_protection_service_policy`.

Figure 8–2 Policy Details Page with the General Tab Selected



Wss10 SAML Token With Message Protection Service Policy

Display Name: Wss10 SAML Token With Message Protection Service Policy

Name: oracle/wss10_saml_token_with_message_protection_service_policy

Category: Security

Description: This policy enforces message-level protection and SAML-based authentication for inbound SOAP requests in accordance with the WS-Security 1.0 standard. It uses WS-Security's Basic 128 suite of asymmetric key technologies, specifically RSA key mechanisms

Enabled: ☑

Attachment Attributes

* Applies To: Service Bindings

Service Category: ● Service Endpoint ○ Client ○ Both

Version Information

Version Number: 1

Last Updated: Monday, December 8, 2014 10:08:45 AM GMT-08:00

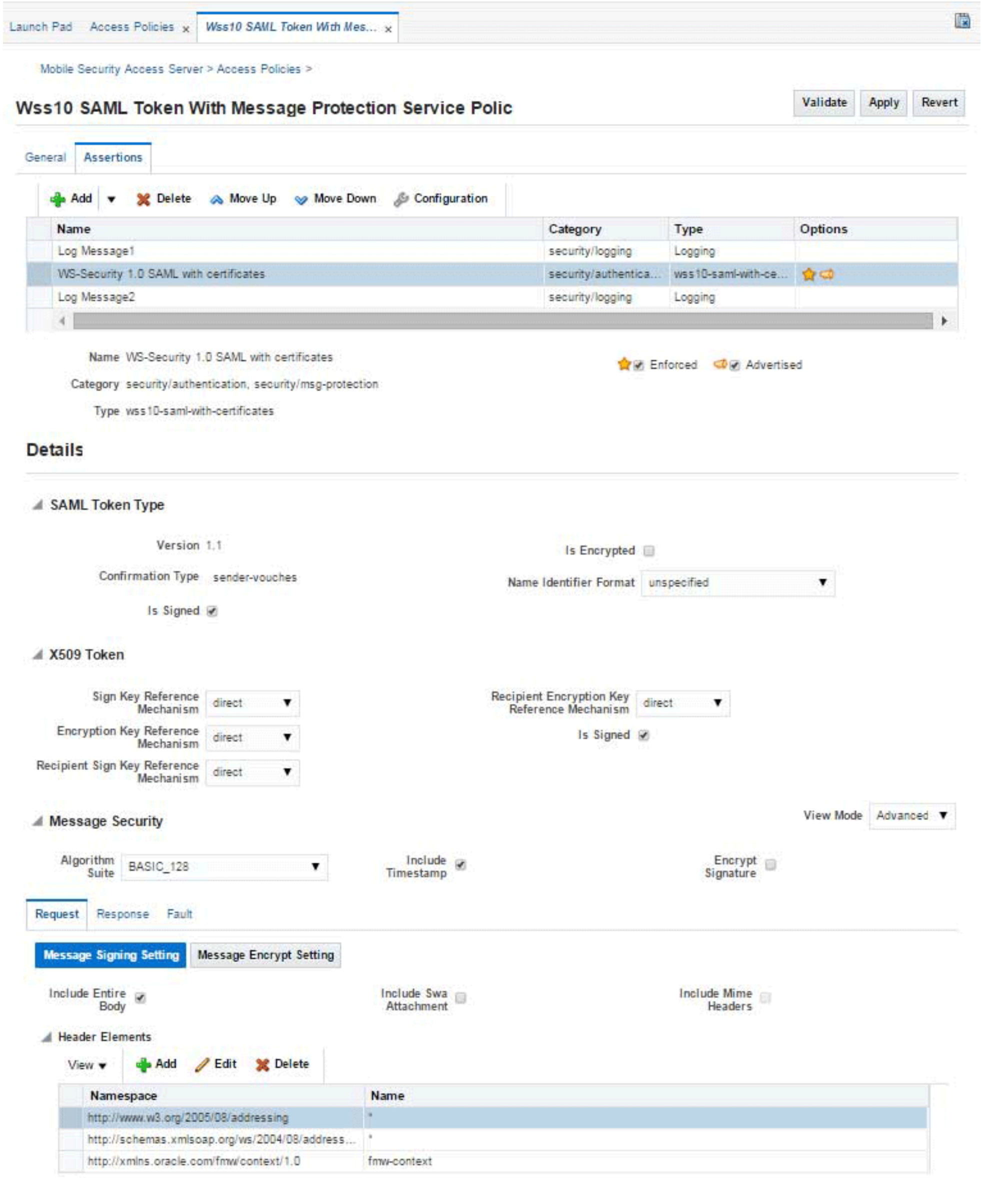
Updated By: default

Copyright © 2000, 2014, Oracle and/or its affiliates. All rights reserved.

The Policy Details page contains two tabs:

- The **General** tab (shown in Figure 8–2) displays information such as the policy name and display name, policy category, description, and whether the policy is enabled. The Attachment Attributes section provides details about the type of endpoints to which the policy can be attached, and the service category (service endpoint, client, or both). The Version Information section lists the version number of the policy, when it was last updated, and by whom. You can navigate to the Policy Version history page. For more information about policy versions, see "Versioning Policies" on page 8-13.
- The **Assertions** tab includes a table that lists all of the assertions contained in the policy. Select the assertion name in the table to view the assertion details. The content displayed varies depending on the assertion selected. Figure 8–3 displays the **Assertions** tab for the Wss10 SAML Token With Message Protection Service Policy.

Figure 8–3 Policy Details Page with the Assertion Tab Selected



Wss10 SAML Token With Message Protection Service Polic

Name	Category	Type	Options
Log Message1	security/logging	Logging	
WS-Security 1.0 SAML with certificates	security/authentica...	wss10-saml-with-ce...	★ ◁
Log Message2	security/logging	Logging	

Name WS-Security 1.0 SAML with certificates

Category security/authentication, security/msg-protection

Type wss10-saml-with-certificates

★ ☑ Enforced ◁ ☑ Advertised

Details

SAML Token Type

Version 1.1

Confirmation Type sender-vouches

Is Signed ☑

Is Encrypted □

Name Identifier Format unspecified

X509 Token

Sign Key Reference Mechanism direct

Encryption Key Reference Mechanism direct

Recipient Sign Key Reference Mechanism direct

Recipient Encryption Key Reference Mechanism direct

Is Signed ☑

Message Security

View Mode Advanced

Algorithm Suite BASIC_128

Include Timestamp ☑

Encrypt Signature □

Include Entire Body ☑

Include Swa Attachment □

Include Mime Headers □

Header Elements

Namespace	Name
http://www.w3.org/2005/08/addressing	*
http://schemas.xmlsoap.org/ws/2004/08/address...	*
http://xmlns.oracle.com/fmw/context/1.0	fmw-context

8.2.4 Creating and Editing a Policy

The following sections describe how to create and edit policies:

- Creating a New Policy
- Cloning a Policy
- Editing a Policy

8.2.4.1 Creating a New Policy

Use the following procedure to create a new policy using one or more assertion templates:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.

2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

3. Click **Create**. Alternatively, select **Actions** and then **Create**.

   The page, temporarily titled Untitled, includes two tabs: **General** and **Assertions**. The **General** tab is displayed by default.

4. On the **General** tab, optionally specify a unique name in the **Display Name** field to be used in the console to reference the policy.

   The page title is updated to reflect the display name you specify. If you do not specify a display name, the policy name is used to reference the policy.

5. Enter a policy name in the **Name** field.

   The policy name must include the directory in which the policy is located. For example, all predefined policies provided by Oracle are contained in the `oracle/` directory, such as `oracle/wss_http_token_service_policy`.

   Notes: Oracle recommends that you follow the policy naming conventions described in "Recommended Naming Conventions for Policies" on page 8-26.

   You cannot edit the name of a policy once the policy is created. To change the policy name, you will need to clone the policy and assign it a different name.

6. By default, the **Category** field is set to Security.

   Note: You can create new policies in the Security category only.

7. Optionally, enter a brief description for the policy in the **Description** field.

8. Select the **Enabled** option to enable the policy, if desired. Note that a policy that is not enabled is not enforced at run time.

9. In the Attachment Attributes section of the page, specify the type of policy enforcement points to which the policy can be attached. From the **Applies To** menu, choose one of the following options:

   - **All**—Specifies that the policy can be attached to any type of policy enforcement point, including service endpoints and client endpoints.
   - **Service Bindings**—Specifies that the policy can be attached to service and client endpoints. When you choose this option, in the **Service Category** field select whether the policy can be attached to service endpoints, service clients, or both.

10. Select the **Assertions** tab, and click **Add** to add assertions to your policy. For more information, see "Adding Assertions to a Policy" on page 8-10

11. Optionally, add an OR group to the policy. Select the **Add** menu then select **OR Group**. Then, select the **Add** menu then select **Assertion to OR Group** to add the desired assertions to the OR group.

    An OR group enables you to define multiple security subcategory options, only one of which can be executed. For example, a subset can contain both a SAML

Token and a Username Token security/authentication subcategory assertion, so an application can use either one or the other, but not both. For more information, see "Adding an OR Group to a Policy" on page 8-12.

12. Configure the assertions as required by modifying the settings and configuration properties.

    To edit the configuration properties, click **Configuration**. The list of configuration properties defined for the assertion are displayed. Edit the configuration properties as described in "Editing the Configuration Properties" on page 8-21 and click **OK**.

    For details about the settings and configuration properties for each assertion template, see "Assertion Template Settings and Configuration Properties" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

13. When you have finished adding assertions to the policy, select the assertions in the table and use the **Move Up** and **Move Down** buttons to set the order in the policy. Assertions are invoked in the order in which they appear in the list.

14. Click **Validate** to validate the policy.

    If the policy is invalid, it is disabled as a precaution. After you correct the validation issues, you will have to enable the policy. For more information on policy validation, see "Validating Policies" on page 8-16.

15. Click **Apply** to apply your changes, or **Revert** to revert your changes.

8.2.4.2 Cloning a Policy

You can create a new policy by cloning an existing policy.

To clone a policy:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.

2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

3. Optionally, refine the list of policies displayed using Search, as described in "Searching for Policies" on page 8-3.

4. Select the policy to be cloned from the list of policies and click **Create Like**. Alternatively, select **Actions** and then **Create Like**.

   > **Notes:** Oracle recommends that you follow the policy naming conventions described in "Recommended Naming Conventions for Policies" on page 8-26.
   >
   > You cannot edit the name of a policy once the policy is created. To change the policy name, you will need to clone the policy and assign it a different name.

5. Modify the policy as required, including the assertions.

   For details about adding assertions to the policy, see "Adding Assertions to a Policy" on page 8-10. For details about adding an OR group to the policy, see "Adding an OR Group to a Policy" on page 8-12.

6. Click **Validate** to validate the policy.

   If the policy is invalid, it is disabled as a precaution. After you correct the validation issues, you will have to enable the policy. For more information on policy validation, see "Validating Policies" on page 8-16.

7. Click **Apply** to apply your changes, or **Revert** to revert your changes.

8.2.4.3 Editing a Policy

> **Note:** Oracle recommends that you do not edit the predefined policies so that you will always have a known set of valid policies.
>
> If you wish to edit a predefined policy, Oracle recommends that you clone the policy and then edit it.

You can edit a policy as described in this section. The changes that you make to the policy take effect at the next polling interval for policy changes.

Each time you save a change to your policy, a new version is created, and the older versions are retained. For more information about policy versioning, see "Versioning Policies" on page 8-13.

To edit a policy:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

3. Optionally, refine the list of policies displayed using Search, as described in "Searching for Policies" on page 8-3.
4. Select the policy to be edited from the list of policies and click **Open**. Alternatively, select **Actions** and then **Open**.

   The Policy Details page is displayed. For more information about the Policy Details page, see "Viewing the Details of a Policy" on page 8-3.

5. Select the **General** tab and edit the following information:
   - Display name and description, if desired. You cannot edit the policy name. To change the name of a policy, you will need to clone it and assign it a different name.
   - Remaining fields on the tab as required, including enabling or disabling the policy or modifying the type of policy enforcement points to which the policy can be attached.
6. Select the **Assertions** tab and perform one or more of the following tasks:
   - Modify the assertion settings and configuration properties as required. To modify the assertion settings, select the assertion in the table and edit the settings as required in the Details section of the page. To edit the configuration properties, click **Configuration** and edit the properties as required in the Configuration table. To enable the assertion, select the **Enforced** option.

     > **Note:** The **Advertised** option is reserved for future use.

- Add assertions or OR groups as required, as described in "Adding Assertions to a Policy" on page 8-10 and "Adding an OR Group to a Policy" on page 8-12, respectively.
- Delete assertions or OR groups as required. To do so, select the assertion or OR group in the table and click **Delete**.

For details about the assertions in each predefined policy, see "Predefined Policies" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

7. Click **Validate** to validate the policy.

   If the policy is invalid, it is disabled as a precaution. After you correct the validation issues, you will have to enable the policy. For more information on policy validation, see "Validating Policies" on page 8-16.

8. Click **Apply** to apply your changes, or **Revert** to revert your changes.

8.2.5 Exporting and Importing Policies

Import and export policies using the procedures described in the following sections.

- Exporting a Policy
- Importing a Policy

8.2.5.1 Exporting a Policy

You may want to export a policy to copy it from a development environment to a production environment, or to simply view the policy in another tool or application. You can export policies that you have created as described in "Creating and Editing a Policy" on page 8-5. Once the policy is exported, you can import it to another repository, attach it, make changes to it, and so forth.

Use the following procedure to export a policy from the repository:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

3. Select the policy or policies to be exported from the list of policies and click **Export**.

   The policies are added to a zip archive file named `policyexport.zip` by default, and downloaded to your local directory.

   If you perform multiple export operations, subsequent files are named uniquely. For example, as `policyexport`*`(n)`*`.zip`, where *n* starts with 1 and is incremented by 1 for each additional export.

   The directory structure for each policy is maintained in the archive file using the following structure:

   `META-INF/policies/`*`policyname`*

8.2.5.2 Importing a Policy

Import one or more policies into the repository using the following procedure. Once the policies are imported, you can attach them and make changes to them.

Notes: The policy name you import must not already exist in the repository.

Be aware that "policy name" and "file name" are different. The policy name is specified by the name attribute of the policy content; the file name is the name of the policy file. You might find it convenient for the two names to match, but it is not required.

You cannot prefix the name of a policy with `oracle_`. Otherwise, you will receive exceptions when you try to use the policy.

To import one or more policies:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Click **Import**.

   You are prompted to provide the name of a zip archive file containing the policies to be imported.

   Note: The policies to be imported must use the following directory structure in the zip archive:

   ```
   META-INF/policies/policyname
   ```

   Within this directory structure, *`policyname`* includes the directory in which the policy is located.

   ---
4. In the Import window, click **Choose File** and navigate to the directory where the policies archive file is located, then select the zip archive file to be imported.
5. Click **Import**.

   If an error is encountered with one of the policies, the import process stops. For example, if there are five policies to be imported and an error is encountered in the third one, the first two will be imported but the remaining policies will not.

   An information window is displayed listing the policies that were imported. Click **OK** to close the window.

   The imported policies are added to the list of policies in the Access Policies page.

8.2.6 Adding Assertions to a Policy

You can add assertions to a policy during policy creation or editing.

The policy can contain any number of assertions belonging to the Security category; however, the combination of assertions must be valid. For more information on valid assertions, see "Validating Policies" on page 8-16.

To add an assertion to a policy:

1. Navigate to the Policy Details page for the policy to which you want to add assertions, as described in "Viewing the Details of a Policy" on page 8-3.

2. Select the **Assertions** tab.
3. Click **Add** or select **Assertion** from the **Add** menu.

   The Add Assertion page is displayed. The assertions available for that policy are displayed in the Search Results table, organized by Template Name. Optionally, use the **View** menu to display the Display Name column, or to change the order of the columns.

4. Select an assertion from the table, or provide search parameters in the **Name** and **Category** fields and click **Search**. The results that match the search criteria are displayed in the Search Results table. In the Search Results table, select the assertion or assertions to be added to the policy and click **Add Selected**. To add all the listed assertions to the policy, click **Add All**.

   The selected assertions are displayed in the Selected Assertion Templates table. The assertions are displayed using the Template Name. Optionally, use the **View** menu to display the Template Display Name column, or to change the order of the columns.

5. In the Selected Assertion Templates table, optionally edit the names for the added assertions in the **Assertion Name** field.
6. Review the selections in the Selected Assertion Templates table. To remove one or more assertions from this table, click **Remove Selected** or **Remove All.** When you have confirmed the assertion selection, click **Add Assertion**.

   The added assertions are listed in a table in the **Assertion** tab.

   For details about the assertion templates, see "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

7. To configure the assertion, select the assertion and edit the settings as required in the Details section of the page.
8. To enable or advertise the assertion, select the **Enforced** option.

   Note: The **Advertised** option is reserved for future use.

9. To edit the configuration properties, click **Configuration**.

   The list of configuration properties defined for the assertion are displayed.

10. Edit the Configuration properties and click **OK**.

    For details about the configuration properties for each assertion template, see "Assertion Template Settings and Configuration Properties" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

    Note that you can edit only the **Value**, and **Description** fields. The **Name**, **Type**, and **Default Value** property settings defined in the assertion template cannot be changed, and are displayed as read only. For details about these properties, see "Editing the Configuration Properties" on page 8-21.

11. When you have finished adding assertions to the policy, select the assertions in the table and use **Move Up** and **Move Down** buttons to set the order in the policy. Assertions are invoked in the order in which they appear in the list.
12. Click **Apply** to apply your changes, or **Revert** to revert your changes.

8.2.7 Adding an OR Group to a Policy

You can create an OR group, consisting of one or more assertions, enabling a single policy to accept multiple types of security tokens. A client can enforce *any one* of the policies that are defined in the OR group. For more information, see "Defining Multiple Policy Alternatives (OR Groups)" on page 8-26.

You can add only one OR group to a policy. Once you have added an OR Group, the **OR Group** option is greyed out.

To add an OR group to a policy:

1. Navigate to the Policy Details page for the policy to which you want to add the OR group.
2. Select the **Assertions** tab.
3. Select **OR Group** from the **Add** menu.

   An `OR Group` row is added to the assertions table.
4. Select **Assertion to OR Group** from the **Add** menu. Notice that the **OR Group** is now greyed out on the menu, so you cannot add any additional OR groups.

   > **Note:** If you click **Add** or select **Assertion** from the **Add** menu, the assertion will be added *outside* the OR group.

   The Add Assertion search page is displayed.
5. Select one or more assertions from the Search Results table, or provide search parameters in the **Name** and **Category** fields and click **Search**. The results that match the search criteria are displayed in the Search Results table.

   For details about the assertion templates, see "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.
6. In the Search Results table, select the assertion or assertions to be added to the OR Group and click **Add Selected**. The selected assertions are displayed in the Selected Assertion Templates table.
7. In the Selected Assertion Templates table, optionally provide display names for the added assertions in the **Assertion Name** field.
8. Review the selections in the Selected Assertion Template table. To remove one or more assertions from this table, click **Remove Selected** or **Remove All.** When you have confirmed the assertion selection, click **Add Assertion**.

   The added assertions are listed under the OR Group in the list of assertions in the **Assertion** tab.
9. To add additional assertions to the OR group, repeat steps 4 through 8.
10. Configure the assertions as required by modifying the settings and configuration properties.
    - To edit the assertion settings, select the assertion and edit the settings in the Details section of the page.
    - To edit the configuration properties, click **Configuration**.

      The list of configuration properties defined for the assertion are displayed.

      Edit the configuration properties as described in "Editing the Configuration Properties" on page 8-21 and click **OK**.

- To enable or advertise the assertion, select the **Enforced** option.

> **Note:** The **Advertised** option is reserved for future use.

For details about the configuration properties for each assertion template, see "Assertion Template Settings and Configuration Properties" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

11. When you have finished adding assertions to the OR group, select the assertions and use **Move Up** and **Move Down** to order them as needed. Assertions are considered for invocation in the order that they appear on the list.
12. To delete an assertion from the OR group, select the assertion and click **Delete**. To delete the entire OR group, select the OR group and click **Delete**.
13. Click **Apply** to apply your changes, or **Revert** to revert your changes.

8.2.8 Versioning Policies

Whenever a change to a policy is saved, a new version of the policy is automatically created and the version number is incremented. The Policy Manager maintains the history of these changes, enabling you to go back to an earlier version.

For example, you might find it useful to create two different versions of a policy and alternate between them. For example, you might have an occasional need to use a policy such as `oracle/binding_authorization_denyall_policy` policy with selected roles to temporarily lock down access to a service.

By using the versioning feature, you can reuse multiple versions of a policy without having to recreate them every time you need them.

You can also delete any version of the policy, except the active policy, from the Policy Version history table by selecting the policy and clicking **Delete**.

You cannot edit the policy from the Policy Version history page. You must edit a policy from the Policy Details page.

The following sections describe versioning in more detail:

- Viewing the Version History of a Policy
- Changing the Current Version of a Policy
- Deleting Versions of a Policy
- Exporting a Version of a Policy

8.2.8.1 Viewing the Version History of a Policy

You can view the version history for a policy from the Policy Version history page, which you can access from the Policy Details page.

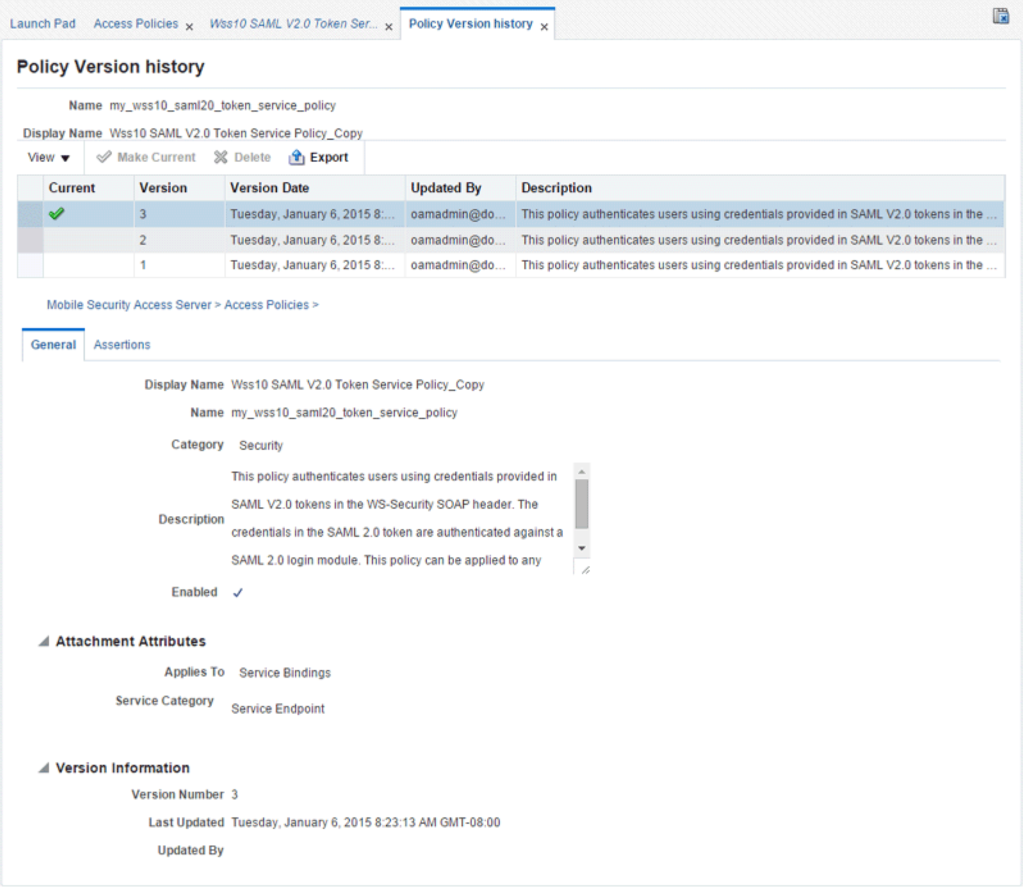
To view the version history for a policy:

1. Navigate to the Policy Details page for the policy as described in "Viewing the Details of a Policy" on page 8-3.
2. Select the **General** tab for the policy, if it is not already selected.
3. In the Version Information section of the page, click **Versioning History**.

   The Policy Version history for the page is displayed, as shown in Figure 8–4. The policy versions appear in order in the version history table at the top of the page.

The currently active policy has the highest version number, and is the only policy that can be attached to a policy enforcement point. However, you can make an earlier version of a policy the active version.

Figure 8–4 Policy Version History Page



Policy Version history

Name my_wss10_saml20_token_service_policy

Display Name Wss10 SAML V2.0 Token Service Policy_Copy

Current	Version	Version Date	Updated By	Description
✓	3	Tuesday, January 6, 2015 8:...	oamadmin@do...	This policy authenticates users using credentials provided in SAML V2.0 tokens in the ...
	2	Tuesday, January 6, 2015 8:...	oamadmin@do...	This policy authenticates users using credentials provided in SAML V2.0 tokens in the ...
	1	Tuesday, January 6, 2015 8:...	oamadmin@do...	This policy authenticates users using credentials provided in SAML V2.0 tokens in the ...

Mobile Security Access Server > Access Policies >

Display Name Wss10 SAML V2.0 Token Service Policy_Copy

Name my_wss10_saml20_token_service_policy

Category Security

Description This policy authenticates users using credentials provided in SAML V2.0 tokens in the WS-Security SOAP header. The credentials in the SAML 2.0 token are authenticated against a SAML 2.0 login module. This policy can be applied to any

Enabled ✓

Attachment Attributes

Applies To Service Bindings

Service Category Service Endpoint

Version Information

Version Number 3

Last Updated Tuesday, January 6, 2015 8:23:13 AM GMT-08:00

Updated By

8.2.8.2 Changing the Current Version of a Policy

Use the following procedure to change the current version of the policy:

1. Navigate to the Policy Details page for the policy as described in "Viewing the Details of a Policy" on page 8-3.
2. Select the **General** tab for the policy, if it is not already selected.
3. In the Version Information section of the policy detail page, click **Versioning History** to display the Policy Version history page.
4. In the policy version table, select the version to be made current and click **Make Current**.

   The selected policy version becomes the current active policy and the current version number is incremented by 1. The earlier version of the policy is retained.

8.2.8.3 Deleting Versions of a Policy

Use the following procedure to delete earlier versions of a policy. You can delete all versions except the active policy version. To delete all versions of the policy, including the active version, see "Deleting a Policy" on page 8-15.

1. Navigate to the Policy Details page for the policy as described in "Viewing the Details of a Policy" on page 8-3.
2. Select the **General** tab for the policy, if it is not already selected.
3. In the Version Information section of the policy detail page, click **Versioning History** to display the Policy Version history page.
4. In the policy version table, select the version or versions to be deleted and click **Delete**.
5. In the Confirm Policy Version Deletion box, click **OK**.

   The selected policy version(s) is deleted from the repository and the Policy History table.

8.2.8.4 Exporting a Version of a Policy

Use the following procedure to export a version of the policy:

1. Navigate to the Policy Details page for the policy as described in "Viewing the Details of a Policy" on page 8-3.
2. Select the **General** tab for the policy, if it is not already selected.
3. In the Version Information section of the policy detail page, click **Versioning History** to display the Policy Version history page.
4. In the policy version table, select the version to be exported and click **Export**.

   The policy is added to a zip archive file named `policyexport.zip` by default, and downloaded to your local directory.

   The directory structure for each policy is maintained in the archive file using the following structure:

   META-INF/policies/*policyname*

8.2.9 Deleting a Policy

Before you delete a policy, Oracle recommends that you verify that the policy is not attached to any URLs. If you try to delete a policy that is attached to a URL, you will receive a warning. You will not be prevented from deleting an attached policy. However, the service request will fail the next time the URL to which the policy is attached is invoked.

When you delete a policy, the active policy and all previous versions of the policy are deleted. To retain the active policy version and delete only the previous versions of the policy, see "Deleting Versions of a Policy" on page 8-14.

To delete a policy:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Optionally, refine the list of policies displayed using Search, as described in "Searching for Policies" on page 8-3.
4. From the Access Policies page, select the policy to be deleted from the list of policies and click **Delete**. Alternatively, select **Actions** and then **Delete**.

5. A dialog box appears asking you to confirm the deletion. Click **Delete**.

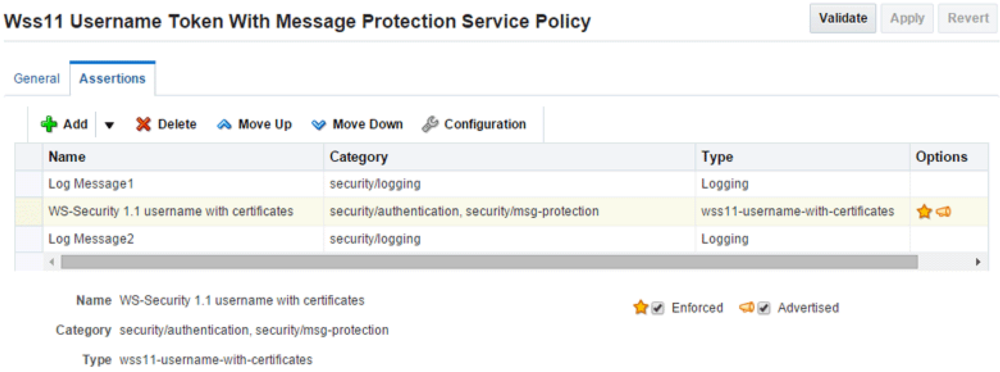
8.3 Validating Policies

There are restrictions on the type and number of policy assertions that are permitted in a policy. A policy can contain only assertions that belong to a single category. Therefore, you cannot combine a security assertion with an management assertion in the same policy. The policy type is determined by the category of the assertion. Therefore, a policy containing a security assertion is a security policy. Security assertions are further categorized into subcategories: authentication, message protection (msg-protection), and authorization.

There are restrictions on the number and type of assertions you can have in a policy. The restrictions are as follows:

- A security policy can contain multiple security assertions; however, there can be only one assertion from the following subcategories in a policy: encryption, signing, and authentication.
- Some assertions contain both authentication and message protection. For example, if you view the `oracle/wss11_username_token_with_message_protection_service_policy`, you will see that the second assertion falls into two categories: security/authentication and security/msg-protection, as shown in Figure 8–5.

Figure 8–5 Security Assertion with Two Subcategories



Name	Category	Type	Options
Log Message1	security/logging	Logging	
WS-Security 1.1 username with certificates	security/authentication, security/msg-protection	wss11-username-with-certificates	★ 📣
Log Message2	security/logging	Logging	

Name WS-Security 1.1 username with certificates

Category security/authentication, security/msg-protection

Type wss11-username-with-certificates

★ ☑ Enforced 📣 ☑ Advertised

Oracle recommends that you create one policy for authentication and message protection, and a second policy for authorization. If you create a policy that contains both an authentication and an authorization assertion, then the authentication assertion must precede the authorization assertion.

When you create a new policy or edit a policy, the validation process checks to see that your policies meet these requirements. If the validation fails during policy creation, the policy is created but is marked as disabled.

To validate a policy:

1. Navigate to the Policy Details page for the policy as described in "Viewing the Details of a Policy" on page 8-3.
2. On the Policy Details page of the policy being viewed or edited, click **Validate**.

   If the validation is successful, the `Policy is Valid` message appears.

   If the validation is not successful, the resulting error message describes the problem. Make the necessary corrections, then revalidate the policy.

3. Once the policy validates successfully, click **Apply** to save the policy, if it is not already saved.

8.4 Managing Policy Assertion Templates

You manage policies from the Access Policies page. From this page you can:

- Search for specific assertion templates
- View assertion templates
- Create, edit, and delete assertion templates
- Edit configuration properties for an assertion template
- Import and export assertion templates

The following sections describe how to manage assertion templates.

- Viewing Assertion Templates
- Searching for an Assertion Template
- Viewing the Details of an Assertion Template
- Cloning an Assertion Template
- Editing an Assertion Template
- Editing the Configuration Properties
- Configuring Assertions
- Exporting and Importing an Assertion Templates
- Deleting an Assertion Template

8.4.1 Viewing Assertion Templates

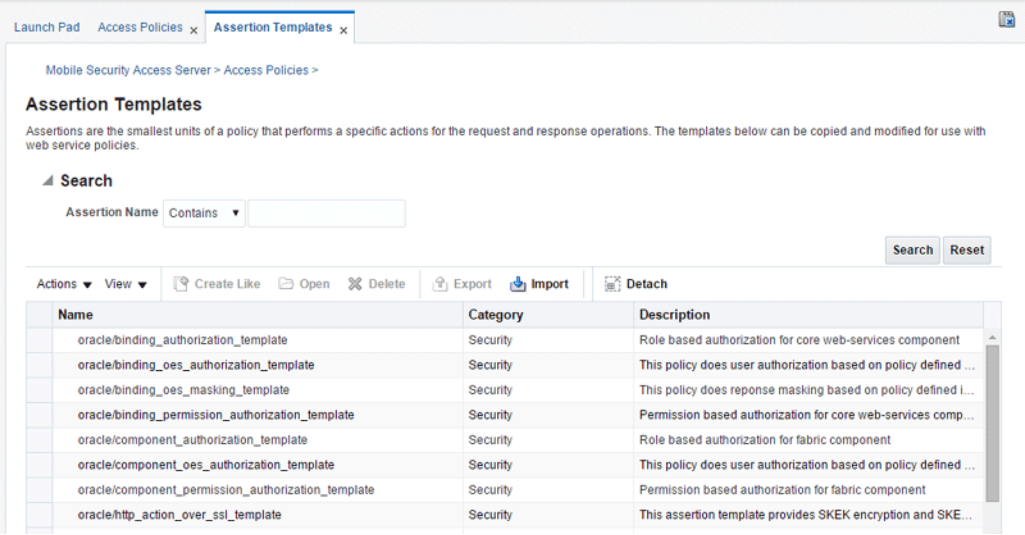
You view assertion templates from the Assertion Templates page by performing the following steps:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.

Figure 8–6 Assertion Templates Page



Assertion Templates

Assertions are the smallest units of a policy that performs a specific actions for the request and response operations. The templates below can be copied and modified for use with web service policies.

Search

Assertion Name: Contains ▾

Name	Category	Description
oracle/binding_authorization_template	Security	Role based authorization for core web-services component
oracle/binding_oes_authorization_template	Security	This policy does user authorization based on policy defined ...
oracle/binding_oes_masking_template	Security	This policy does reponse masking based on policy defined i...
oracle/binding_permission_authorization_template	Security	Permission based authorization for core web-services comp...
oracle/component_authorization_template	Security	Role based authorization for fabric component
oracle/component_oes_authorization_template	Security	This policy does user authorization based on policy defined ...
oracle/component_permission_authorization_template	Security	Permission based authorization for fabric component
oracle/http_action_over_ssl_template	Security	This assertion template provides SKEK encryption and SKE...

8.4.2 Searching for an Assertion Template

In the Assertion Templates page, you can reduce the number of assertion templates that are returned by specifying the appropriate search criteria. To do so:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.
4. In the Search pane, specify the criteria to use in the search.

   In the **Assertion Name** field, enter an assertion template name or part of a name and select the operator to use to refine the search. Available operators are Starts with, Ends with, Equals, and Contains. For example, to search for message protection assertion templates only, select the **Contains** operator, and enter `message` in the **Assertion Name** field.

   You can use percent `%` as a wildcard, any place in the name. Asterisk `*` is not recognized as a wildcard and is treated as plain text. Searches are case-insensitive.
5. Click **Search**.

   The Assertion Templates table is refreshed to include only those assertion templates that match the specified search criteria.

8.4.3 Viewing the Details of an Assertion Template

Use the following procedure to view the details of an assertion template.

To view the assertion template details:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.

2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

3. Click **Assertion Templates**.

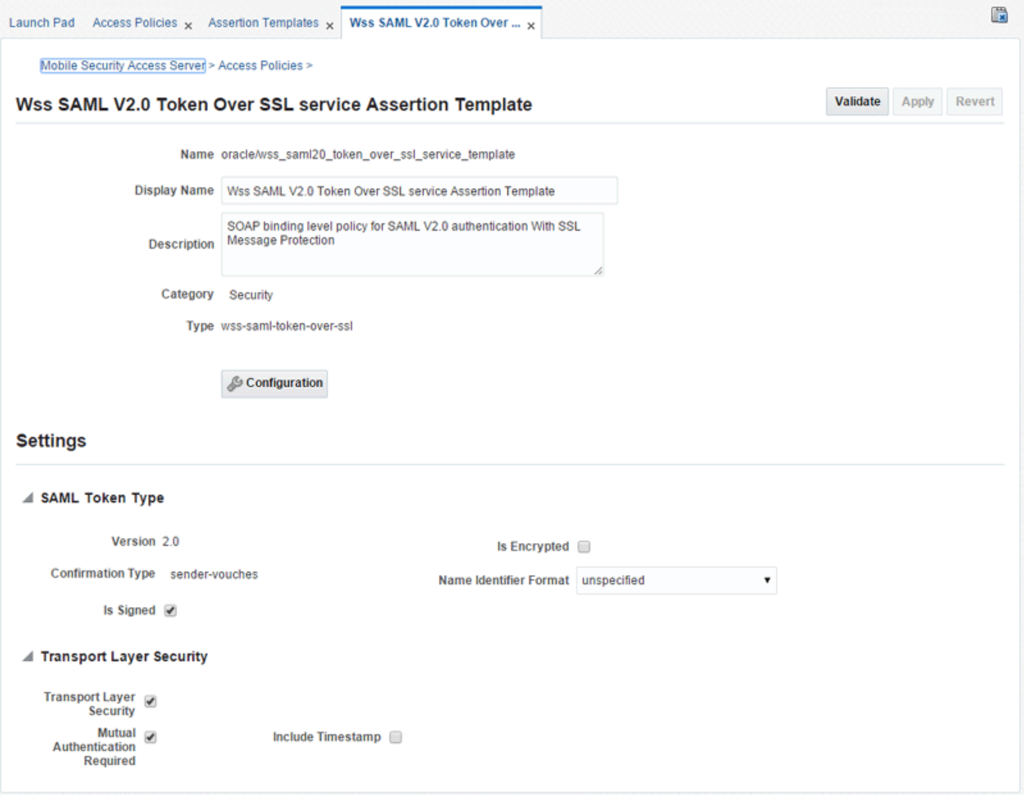
   The Assertion Templates page opens in a new tab.

4. Optionally, refine the list of assertion templates displayed using Search, as described in "Searching for an Assertion Template" on page 8-18.

5. Select the assertion template to be viewed from the list of assertion templates and click **Open**. Alternatively, select **Actions** and then **Open**.

   Figure 8–7 displays the Assertion Template Details page for the Wss10 SAML V2.0 Token with Message Protection service Assertion Template.

Figure 8–7 Assertion Template Details Page



Wss SAML V2.0 Token Over SSL service Assertion Template

Name oracle/wss_saml20_token_over_ssl_service_template

Display Name Wss SAML V2.0 Token Over SSL service Assertion Template

Description SOAP binding level policy for SAML V2.0 authentication With SSL Message Protection

Category Security

Type wss-saml-token-over-ssl

Settings

SAML Token Type

Version 2.0

Is Encrypted ☐

Confirmation Type sender-vouches

Name Identifier Format unspecified

Is Signed ☑

Transport Layer Security

Transport Layer Security ☑

Mutual Authentication Required ☑

Include Timestamp ☐

6. Review the details of the assertion template.

   General information about the assertion template is provided at the top of the page. Click **Configuration** to view the configuration properties for the template. The Settings section of the page displays the settings specific to that template. For details about the settings and configuration properties for each of the predefined assertion templates, see "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

8.4.4 Cloning an Assertion Template

You can create a new assertion template using an existing template as the base. Select the assertion template that most closely matches the desired behavior, make a copy of it using the **Create Like** feature, then make any changes required to get the new behavior.

To clone a policy:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.
4. Optionally, refine the list of assertion templates displayed using Search, as described in "Searching for an Assertion Template" on page 8-18.
5. Select the assertion template to be cloned from the list of assertion templates and click **Create Like**. Alternatively, select **Actions** and then **Create Like**.

   The Assertion Template Details page is displayed.
6. Edit the name and display name for the assertion template and, optionally, enter a brief description.

   The word *Copy* is appended to the name and display name of the cloned assertion template and, by default, this is the name assigned to the new assertion template.

   It is recommended that you change the name of this new assertion template to be more meaningful in your environment. For more information, see "Recommended Naming Conventions for Assertion Templates" on page 8-28.

   > **Notes:** You cannot edit the name of an assertion template after it is created. To change the assertion template name, you will need to clone the assertion template and assign it a different name.

7. Modify the assertion template settings and configuration properties as required. For details about the settings and configuration properties in each of the predefined assertion templates, see "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*. For details about modifying the configuration properties, see "Editing the Configuration Properties" on page 8-21.
8. Click **Apply** to save the new assertion template.

8.4.5 Editing an Assertion Template

> **Note:** Oracle recommends that you do not edit the predefined assertion templates so that you will always have a known set of valid templates.
>
> If you wish to edit a predefined assertion template, Oracle recommends that you clone the assertion template and then edit it.

You can edit an assertion template as described in the following procedure.

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.
4. Optionally, refine the list of assertion templates displayed using Search, as described in "Searching for an Assertion Template" on page 8-18.
5. Select the assertion template to be edited from the list of assertion templates and click **Open**. Alternatively, select **Actions** and then **Open**.
6. Edit the display name and description, if desired. You cannot edit the assertion template name. To change the name of an assertion template you will need to clone it and assign it a different name.
7. Edit the settings as required.

   For details about the settings and configuration properties for each of the predefined assertion templates, see "Predefined Assertion Templates" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.
8. Click **Configuration** to edit the configuration properties.

   To delete a property, select the property in the table and click **Delete**.
9. Click **OK** to accept the configuration property changes.
10. Click **Apply** to apply your changes, or **Revert** to revert your changes.

8.4.6 Editing the Configuration Properties

If you have cloned one of the predefined assertion templates, you can modify the configuration properties to match your environment. For example, properties that are configurable in assertion templates include `csf-key`, `saml.issuer.name`, `keystore.recipient.alias`, and `role`, among others.

When you clone an assertion template, or edit a cloned assertion template, you can configure the following settings for each property:

- **Description**—Description of the property.
- **Value**—Current value.
- **Default**—Default value. This value is used if the **Value** field is not set.
- **Type**—Can be one of the following:
  - **Constant**—Property cannot be overridden.
  - **Required**—Property is required and can be overridden.
  - **Optional**—Property is optional and can be overridden.

To configure the properties:

1. In the assertion template being cloned or edited, click **Configuration**.

   The Configuration window displays the list of properties for the template.
2. Select the property from the list and modify the fields as required. Note that the Name of an existing property cannot be changed.

3. Add or delete configuration properties as required.

   To add a configuration property, click **Add**. In the blank row that appears, provide a name for the property. The remaining fields are optional. However, if you select Type **required**, then you must provide a value for the property.

   To delete a configuration property, select the property in the table and click **Delete**.

4. When you have finished changing the configuration properties, click **OK**.
5. Click **Apply** to apply your changes, or **Revert** to revert your changes.

> **Note:** When you add an assertion to a policy, as described in "Adding Assertions to a Policy" on page 8-10, you can modify the **Value**, **Default**, and **Description** configuration properties to match your environment. The **Name** and **Type** configuration properties defined in the assertion template cannot be changed, and are not editable fields in the table.

8.4.7 Configuring Assertions

You can modify the configuration properties to match your environment. For example, properties that are configurable in assertion templates include `csf-key`, `saml.issuer.name`, `keystore.recipient.alias`, and `role`, among others.

When you clone or edit an assertion template, you can configure the following settings for each property:

- **Value**—Current value.
- **Default**—Default value. This value is used if the **Value** field is not set.
- **Type**—Can be one of the following:
  - **constant**—Property cannot be overridden.
  - **required**—Property is required and can be overridden.
  - **optional**—Property is optional and can be overridden.
- **Description**—Description of the property.

To configure the properties:

1. In the assertion template being cloned or edited, click **Configuration**.

   The Configuration window displays the list of properties for the template.

2. Select the property from the list and modify the fields as required. Note that the Name of an existing property cannot be changed.
3. Add or delete configuration properties as required.

   To add a configuration property, click **Add**. In the blank row that appears, provide a name for the property. The remaining fields are optional. However, if you select Type **required**, then you must provide a value for the property.

   To delete a configuration property, select the property in the table and click **Delete**.

4. When you have finished changing the configuration properties, click **OK**.
5. Click **Apply** to save the changes in the assertion template.

> **Note:** When you add an assertion to a policy, as described in "Adding Assertions to a Policy" on page 8-10, you can modify the **Value**, **Default**, and **Description** configuration properties to match your environment. The **Name** and **Type** configuration properties defined in the assertion template cannot be changed, and are not editable fields in the table.

8.4.8 Exporting and Importing an Assertion Templates

Export and import an assertion template, as described in the following sections:

- "Exporting an Assertion Template" on page 8-23
- "Importing an Assertion Template" on page 8-23

8.4.8.1 Exporting an Assertion Template

To export one or more assertion templates:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.
4. Select the assertion templates that you would like to export in the Assertion Templates table.
5. Click **Export**.

   The assertion templates are added to a zip archive file named `assertiontemplatesexport.zip` by default, and downloaded to your local directory.

   If you perform multiple export operations, subsequent files are named uniquely. For example, as `assertiontemplatesexport`*`(n)`*`.zip`, where *`n`* starts with 1 and is incremented by 1 for each additional export.

8.4.8.2 Importing an Assertion Template

Import an assertion template into the repository using the following procedure. Once the assertion template is imported, you can edit it, add it to a policy, and so on.

> **Notes:** The assertion template you import must not already exist in the repository. Otherwise, you will get an error and the import operation will fail.

To import an assertion template:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

The Access Policies page opens in a new tab.

3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.

4. Click **Import**.

   You are prompted to provide the name of a zip archive file containing the assertion templates to be imported.

5. In the Import window, click **Choose File** and navigate to the directory where the assertion template archive file is located, then select the zip archive file to be imported.

6. Click **Import**.

   An information window is displayed listing the policies that were imported. Click **OK** to close the window.

   The imported policies are added to the list of policies in the Access Policies page.

8.4.9 Deleting an Assertion Template

Follow the steps in this section to delete an assertion template that you created or imported. The predefined assertion templates delivered with OWSM are read-only and cannot be deleted.

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.

2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.

3. Click **Assertion Templates**.

   The Assertion Templates page opens in a new tab.

4. Optionally, refine the list of assertion templates displayed using Search, as described in "Searching for an Assertion Template" on page 8-18.

5. Select the assertion template to be deleted from the list of assertion templates and click **Delete**.

   You are prompted to confirm that you want to delete the assertion template.

6. Confirm your selection and click **Delete**.

   The selected assertion template is deleted from the list of assertion templates on the Assertion Templates page.

8.5 Enabling or Disabling Policies and Assertions

The following sections describe the different methods for enabling or disabling policies, or assertions within a policy:

- Enabling or Disabling a Policy for all Policy Enforcement Points
- Enabling or Disabling Assertions Within a Policy

8.5.1 Enabling or Disabling a Policy for all Policy Enforcement Points

When you create a policy, it is enabled by default unless it has validation errors. A policy can be globally enabled or disabled from the Policy Details page. You can enable or disable the policy from one central location, and it will be enabled or disabled for any policy enforcement point to which it is attached.

When you disable a policy from the Policy Details page, the policy continues to be attached to the policy enforcement points, but the policy is not enforced. You may want to temporarily disable a policy if you discover that there is a problem with the policy that is causing all requests to a service to fail. Once the problem is corrected, you can globally enable the policy.

To enable or disable a policy for all policy enforcement points:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Optionally, refine the list of policies displayed using Search, as described in "Searching for Policies" on page 8-3.
4. Select the policy to be edited from the list of policies and click **Open**. Alternatively, select **Actions** and then **Open**.

   The Policy Details page is displayed. For more information, see "Viewing the Details of a Policy" on page 8-3.
5. Select the General tab if it is not already selected.
6. Select or deselect the **Enabled** box to enable or disable the policy, respectively.
7. Click **Save**.

8.5.2 Enabling or Disabling Assertions Within a Policy

Rather than enable or disable an entire policy as described in "Enabling or Disabling a Policy for all Policy Enforcement Points" on page 8-25, you may wish to enable or disable one or more of the assertions that are contained within a policy. This provides a more fine-grained level of control over the assertions that are executed.

To enable or disable one or more assertions within a policy:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Access Policies** in the Mobile Security Access Server section.

   The Access Policies page opens in a new tab.
3. Optionally, refine the list of policies displayed using Search, as described in "Searching for Policies" on page 8-3.
4. Select the policy to be edited from the list of policies and click **Open**. Alternatively, select **Actions** and then **Open**.

   The Policy Details page is displayed. For more information, see "Viewing the Details of a Policy" on page 8-3.
5. Select the **Assertions** tab.

6. Select the assertion in the table and select or deselect the **Enforced** box to enable or disable the assertion within the policy, respectively.
7. Click **Apply**.

8.6 Defining Multiple Policy Alternatives (OR Groups)

To define multiple alternatives for policy enforcement, you can define a set of assertions, called an **OR group**, within a service policy. At run time, based on the assertions defined in the OR group on the service side, a client has the flexibility to choose which *one* of the assertions to enforce.

For example, if a service-side policy defines an OR group that consists of the following assertions:

- `wss11-saml-with-certificates`
- `wss11-username-with-certificates`

At run-time, the client can choose to enforce either the `wss11-saml-with certificates` assertion OR `wss11-username-with-certificates` assertion.

There is no limit to the number of assertions that can be included in an OR group. Each assertion must be valid for the policy and should support the policy requirements.

When defining the OR group, carefully consider the order in which the assertions are added and the settings that are configured. For example, consider the following scenario:

- On the client side, you have attached the `wss11_username_token_with_message_protection_client_policy` policy with `Include Timestamp` enabled.
- On the service side, you have attached a custom OR group policy with two `wss11_username_token_with_message_protection_service_template` assertions defined, the first with `Include Timestamp` disabled and the second with `Include Timestamp` enabled.

In this scenario, the first assertion will get executed and the response will be sent with no timestamp. As a result, processing on the client side will fail because it is expecting a timestamp. This type of situation can occur whenever a client policy assertion expects a greater number of security requirements than the executed service policy assertion.

The following predefined client policies contain OR groups:

- `oracle/multi_token_client_policy`—For more information, see "oracle/multi_token_client_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.
- `oracle/multi_token_over_ssl_client_policy`—For more information, see "oracle/multi_token_over_ssl_client_policy" in *Policy and Assertion Template Reference for Oracle Mobile Security Access Server*.

8.7 Recommended Naming Conventions for Policies

The valid characters for policy names are:

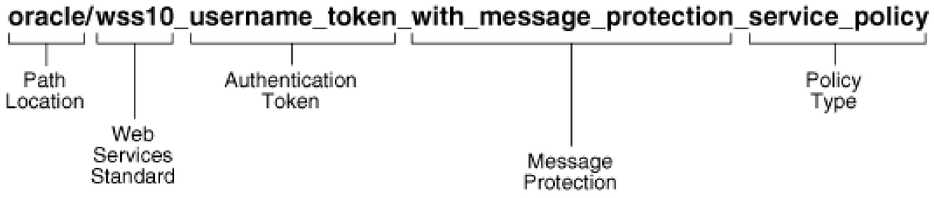
- Uppercase and lowercase letters
- Numerals
- Underscore (_)

- Hyphen (-)

> **Note:** The first character in the name cannot be a hyphen.

Oracle recommends that you encode as much information as possible into the full policy name and display name so that you can tell, at a glance, what the policy does. For example, the full name for one of the predefined security policies is `oracle/wss10_username_token_with_message_protection_service_policy` and the display name is Wss11 Username Token With Message Protection Service Policy. Figure 8–8 identifies the different parts of this predefined policy name.

Figure 8–8 Identifying the Different Parts of a Policy Name



The following convention is used to name the predefined policies. The parts of the policy name are separated with an underscore character (_).

- Path Location – All policies are identified by the directory in which the policy is located. All predefined policies are in the `oracle` directory. Oracle recommends that you keep any policies that you create in a directory that is separate from the `oracle` directory in which the predefined policies are located.
- Web services Standard – If the policy uses a WS-Security standard, it is identified with wss10 (WS-Security 1.0) or wss11 (WS-Security 1.1). Or it could just be set to indicate that it is independent of WS-Security 1.0 or 1.1.
- Authentication token – If the policy authenticates users, then the type of token is specified. The predefined options include:
  - http_token – HTTP token
  - jwt_token - JWT token
  - kerberos_token – Kerberos token
  - saml_token – SAML token
  - saml_hok_token - SAML holder of key token
  - saml20_token - SAML 2.0 token
  - saml20_token_bearer - SAML Bearer 2.0 token
  - username_token – Username and password token
  - x509_token – X.509 certificate token

  You can also define custom authentication tokens.
- Transport security – If the policy requires that the message be sent over a secure transport layer, then the token name is followed by *over_ssl*, for example, `oracle/http_oauth2_mobile_client_over_ssl_policy`.
- Message protection – If the policy also provides message confidentiality and message integrity, then this is indicated using the phrase *with_message_protection* as

in Figure 8–8.

- Policy Type – Indicates the type of policy or assertion template— *client* or *service*. Use the term *policy* to indicate that it is a policy, or *template* to indicate that it is an assertion template. For example, there are predefined policy and template assertions that are distinguished, as follows:

  `wss10_message_protection_service_policy`

  `wss10_message_protection_service_template`

Whatever conventions you adopt, Oracle recommends you take some time to consider how to name your policies. This will make it easier for you to keep track of your policies as your enterprise grows and you create new policies.

It is recommended that you keep any policies you create in a directory that is separate from the oracle directory where the predefined policies are located. You can organize your policies at the root level, in a directory other than oracle, or in subdirectories. For example, all of the following are valid:

- `wss10_message_protection_service_policy`
- `oracle/hq/wss10_message_protection_service_policy`
- `hq/wss10_message_protection_service_policy`

> **Note:** Use of the prefix "`oracle_`" in the policy name (for example, `oracle_wss_http_token_service_policy`) is not recommended as a best practice.

8.8 Recommended Naming Conventions for Assertion Templates

The same naming conventions used to name predefined policies are used to name the assertion templates. The predefined assertion templates begin with the directory name `oracle/` and are identified with the suffix `_template` at the end; for example, `oracle/wss10_message_protection_service_template.`

It is recommended that you follow the recommended naming conventions, and keep any assertion templates that you create in a directory that is separate from the oracle directory where the predefined assertion templates are located. You can organize your assertion templates at the root level, in a directory other than oracle, or in subdirectories.

For more information about the naming conventions for predefined policies, see "Recommended Naming Conventions for Policies" on page 8-26.

9 Managing Log Files

Mobile Security Access Server (MSAS) components generate log files containing messages that record all types of events. This chapter describes how to view and manage log files to assist in monitoring system activity and in diagnosing problems.

It contains the following sections:

- Overview of Log File Management
- Configuring the Level of Information Written to Log Files
- Configuring MSAS Access Logs

9.1 Overview of Log File Management

Mobile Security Access Server (MSAS) components generate log messages that record all types of events, including startup and shutdown information, errors, warning messages, and access information on HTTP requests.

To capture log messages, loggers are provided for each MSAS component, as described in Table 9–1.

Table 9–1 Loggers for MSAS Components

Logger Name	MSAS Component
`oracle.idm.gateway.common`	MSAS common libraries
`oracle.idm.gateway.gmsclient`	MSAS management client
`oracle.idm.gateway.grs`	MSAS run-time server
`oracle.idm.gateway.snapshot`	MSAS security artifacts snapshot manager
`oracle.security.jps`	OPSS libraries
`oracle.wsm`	Oracle Web Services Manager (Oracle WSM) run-time libraries
`com.sun.jersey`	Jersey JAX-RS

Table 9–2 defines the valid logging levels that you can configure for each MSAS component logger defined in the previous table.The log configuration applies to the logical MSAS instance, and is used by all physical MSAS instances to which the logical instance is bound.

Table 9–2 Logging Level Values

Logging Level	Description
SEVERE	Serious problem that requires immediate attention from the administrator and is not caused by a bug in the product.
WARNING	Potential problem that should be reviewed by the administrator.
INFO	Major lifecycle event such as the activation or deactivation of a primary sub-component or feature.
CONFIG	Configuration information to assist in debugging problems that may be associated with particular configurations
FINE	Detailed tracing messages that can cause a small performance impact. You can enable this level occasionally on a production environment to debug problems.
FINER	Fairly detailed tracing messages that can cause a high performance impact. This level should ***not*** be enabled on a production environment, except on special situations to debug problems.
FINEST	Highly detailed tracing messaged that can cause a very high performance impact. This level should ***not*** be enabled in a production environment. It is intended to be used to debug the product on a test or development environment.

MSAS log files are stored in the following directory:

- `instance_root/instance_name/log`

In this pathname, *instance_root* is the root directory you specified when you created the instance, and *instance_name* is the name of the instance. By default, *instance_root* is *MW_HOME*/instances, and *MW_HOME* is the Middleware home directory in which you installed Mobile Security Access Server.

The current MSAS log file is named as follows: *instance_root*/*instance_name*-diagnostic-0.log. For example: MSAS-123456-diagnostic-0.log. When the current log file reaches 10 MB, it will be backed up to a file using the following format: *instance_root*/*instance_name*-diagnostic-*n*.log, where *n* starts at 1 and is incremented by 1 for each additional log file that is backed up.

9.2 Configuring the Level of Information Written to Log Files

You can configure the message levels using MSAS Console or WLST commands, as described in the following topics:

- Configuring Log Levels Using the MSAS Console
- Configuring Log Levels Using WLST

9.2.1 Configuring Log Levels Using the MSAS Console

To configure the log levels using the MSAS console pages:

1. From the Oracle Access Management home page, click the **Mobile Security** tab from the list of tabs at the top of the page.
2. From the Mobile Security Launch Pad, click **Environments** in the Mobile Security Access Server section.

   The Environments page opens in a new tab.
3. Click **Instances** in the MSAS Environment section.

   The MSAS Instances page opens in a new tab.

4. Click **Configure** for the MSAS instance that you want to configure.

   The MSAS Instance Configuration page opens in a new tab.

5. Click **System Settings** and expand the **Log Configuration** section.

   The list of loggers and associated logging levels are shown in the table. Optionally, use the **View** menu to change the display or order of the columns.

6. To configure the log level for a MSAS component logger:

   a. If the logger does not exist in the table, click **Add** to add a new row to the table and enter the name of the logger in the Logger Name row. For a list of valid loggers, see Table 9–1.

   b. Select a log level for the logger from the drop down list.

7. To delete a logger configuration, select the logger in the table and click **Remove**.

8. Click **Apply** to save the configuration changes or **Revert** to revert them.

9.2.2 Configuring Log Levels Using WLST

You can use WLST commands to configure log levels for the MSAS components. Specifically, you can perform the following configuration tasks:

- Get the message level for a specific logger using the `getMSASLogLevel()` command, as described in "Getting the Log Level Using WLST" on page 9-3.
- Set the message level for a logger using the `setMSASLogLevel()` command, as described in "Setting the Log Level Using WLST" on page 9-4.
- List the configured message levels for all loggers using the `listMSASLoggers()` command, as described in "Getting a List of Loggers" on page 9-4.

You must connect to the Mobile Security Manager Administration Server before you can use the MSAS management commands. For more information, see "Accessing the MSAS WLST Commands" on page 6-27.

9.2.2.1 Getting the Log Level Using WLST

You can view the log level for an MSAS component using the `getMSASLogLevel` WLST command.

For example, to view the log level for the MSAS common libraries, use the following command:

```
getMSASLogLevel('myMSASInstance','oracle.idm.gateway.common')

SEVERE
```

In this example, the log level for the `oracle.idm.gateway.common` logger is set to `SEVERE`.

In the following example, the MSAS run-time server Logger `oracle.idm.gateway.grs` is not configured for the MSAS instance `myMSASInstance`. In this case, it will inherit the configuration from parent logger (`<root>`).

```
getMSASLogLevel('myMSASInstance','oracle.idm.gateway.grs')

Logger "oracle.idm.gateway.grs" is not configured for the MSAS instance
"myMSASInstance". Hence it will inherit the configuration from parent logger.
```

9.2.2.2 Setting the Log Level Using WLST

You can set the log level for an MSAS component using the `setMSASLogLevel` WLST command. For example, to set the log level for the MSAS run-time server, use the following command:

```
setMSASLogLevel('myMSASInstance', 'oracle.idm.gateway.grs', 'WARNING')

Logging configuration for the MSAS instance "myMSASInstance" updated successfully.
```

9.2.2.3 Getting a List of Loggers

To get a list of loggers that have been configured for the MSAS instance `myMSASInstance`, use the following `listMSASLoggers` command, as follows:

```
listMSASLoggers('myMSASInstance')

--------------------------------------------------+-----------------
Logger                                            | Level
--------------------------------------------------+-----------------
<root>                                            | INFO
oracle.idm.gatewya.grs                            | FINEST
oracle.wsm                                        | SEVERE
```

9.3 Configuring MSAS Access Logs

The MSAS access log records all requests processed by the server and is stored in the standard log directory using the name `access.log`:

```
instance_root/instance_name/log/access.log
```

Access logs are enabled by default, but you can disable them if desired using the following configuration property:

- Category: `ServerSettings`
- Property Name: `access.log.enabled`

For details about setting this property, see "Configuring Access Log Settings Using WLST" on page 6-54.

The access log uses the format:

```
%h %u %t \"%r\" %{X-Original-Scheme}i %{X-Original-URL}i %s %b %D
```

where:

- `%h`—Remote host
- `%u`—Remote user
- `%t`—Time the request was received (standard english format)
- `%r`—First line of request
- `%{X-Original-Scheme}i`—Value of X-Original-Scheme header (if present)
- `%{X-Original-URL}i`—Value of X-Original-URL header (if present)
- `%s` - Status of the request
- `%b` - Size of response in bytes, excluding HTTP headers
- `%D` - The time taken to serve the request, in microseconds.

Note: The format of the access.log file is based on the Apache Module mod_log_config file format at `http://httpd.apache.org/docs/2.2/mod/mod_log_config.html`.

10

Managing the MSAS Repository

The following topics provide guidance for maintaining the Mobile Security Access Server (MSAS) repository:

- About the MSAS Repository
- Understanding the Different Mechanisms for Exporting and Importing Application Metadata
- Exporting and Importing MSAS Application Metadata Using WLST
- Migrating MSAS Application Metadata Between Application Environments
- Replacing the MSAS Application Host and Port Values

10.1 About the MSAS Repository

Mobile Security Access Server uses an MDS repository to store MSAS application metadata, including policies, assertion templates, schemas, and so on.

For policies stored within the MSAS repository, each policy has a URI that is evaluated to form a path in which to locate a particular XML document containing the policy. MSAS does not use the MDS customization feature, so all policies are stored as complete documents. Although MDS supports the ability to store multiple versions of a given document, MSAS only accesses the latest version during policy enforcement.

10.2 Understanding the Different Mechanisms for Exporting and Importing Application Metadata

You can use the MSAS console pages in the OAM console or WebLogic Scripting Tool (WLST) commands to export and import application metadata from and to the MSAS repository.

You can use the MSAS console to selectively export and import one policy at a time. The procedures for exporting and importing policies using the MSAS console are described in the following sections:

- "Exporting a Policy" on page 8-9
- "Importing a Policy" on page 8-9

You can export and import MSAS applications using the MSAS console as described in the following sections:

- "Exporting MSAS Applications" on page 3-10
- "Importing MSAS Applications" on page 3-10

You can also use the WLST commands exportMSASAppMetadata and importMSASAppMetadata to facilitate exporting and importing MSAS application metadata directly from and to the MSAS repository. For details about using these commands, see "Exporting and Importing MSAS Application Metadata Using WLST" on page 10-2.

10.3 Exporting and Importing MSAS Application Metadata Using WLST

You can export and import MSAS applications to and from the MSAS repository using the exportMSASAppMetadata and importMSASAppMetadata WLST commands as described in the following sections:

- Exporting MSAS Applications from the MSAS Repository
- Importing MSAS Application Metadata from the MSAS Repository

For more information about the WLST commands and their arguments, see "Repository Commands" in *WebLogic Scripting Tool Command Reference for Identity and Access Management*.

10.3.1 Exporting MSAS Applications from the MSAS Repository

To export MSAS applications from the MSAS repository to a supported ZIP archive file, use the exportMSASAppMetadata command.

```
exportMSASAppMetadata(archiveFileName,[instanceName=None],[applicationName=None],[
includeShared='false'])
```

Note the following:

- If the archive specified using the archiveFileName argument already exists, you can choose to merge the content into the existing archive, overwrite the existing archive, or cancel the operation. If you choose the overwrite option, the original archive is backed up and a message describes the location of the backup archive.

  For example, the following command exports the metadata for all the MSAS applications across all MSAS instances into an archive named MSASApplications.zip.

  ```
  wls:/mydomain/serverConfig> exportMSASAppMetadata('/tmp/MSASApplications.zip')
  ```

- Use the optional instanceName argument to specify the name of the MSAS instance from which you want to export the metadata of one or more MSAS applications. The MSAS instance name is case-sensitive.

  For example, the following command exports the metadata for all applications across all MSAS instances that begin with MSAS into the MSASApplications.zip archive.

  ```
  wls:/mydomain/serverConfig>
  exportMSASAppMetadata('/tmp/MSASApplications.zip','MSAS%')
  ```

- Use the optional applicationName argument to specify the name of the MSAS applications for which you want to export the metadata. The application name is not case-sensitive.

  For example, the following command exports the metadata for all applications that begin with virtual across all MSAS instances into the MSASApplications.zip archive.

  ```
  wls:/mydomain/serverConfig>
  ```

```
exportMSASAppMetadata('/tmp/MSASApplications.zip','',['virtual%'])
```

- If no MSAS instance name is specified, but a valid MSAS application name is specified, then the metadata for all MSAS instances containing this MSAS application is exported. When neither the MSAS instance nor MSAS application name is specified, then the metadata for all the MSAS applications across all MSAS instances is exported.

  For example, the following command exports the metadata for all applications that begin with `virtual` across all MSAS instances into the `MSASApplications.zip` archive.

  ```
  wls:/mydomain/serverConfig>
  exportMSASAppMetadata('/tmp/MSASApplications.zip','',['virtual%'])
  ```

- Use the `includeShared` argument to specify whether the shared documents (access policies referenced by MSAS applications) should be included in the export.

  For example, the following command exports the metadata for all applications that begin with `virtual`, including the application's shared resources, on the `MSAS-123456` instance into the `MSASApplications.zip` archive.

  ```
  wls:/mydomain/serverConfig>
  exportMSASAppMetadata('/tmp/MSASApplications.zip','MSAS-1234561',['virtual%'],t
  rue)
  ```

10.3.2 Importing MSAS Application Metadata from the MSAS Repository

To import MSAS application metadata from the MSAS repository to a supported ZIP archive file, use the `importMSASAppMetadata` command.

```
importMSASAppMetadata(archiveFileName,[instanceName=None],[applicationName=None],[
includeShared='false'])
```

Note the following:

- Use the optional `mapFileName` argument to specify the location of an input map file that describes how to map physical information from the source environment to the target environment.

  For example, the following command imports application metadata from the specified `MSASartifacts.zip` archive according to the `MSASmapfile.txt` map file.

  ```
  wls:/mydomain/serverConfig>
  importMSASAppMetadata('/tmp/MSASartifacts.zip','/tmp/MSASmapfile.txt')
  ```

- You can generate a new map file by setting the `generateMapFlag` argument to `true`. If the map file already exists, it will be overwritten; however, no MSAS application metadata is modified when this argument is set to `true`, only a map file is generated. If you specify a map file without setting the `generateMapFlag`, or setting it to `false`, and the map file does not exist, the operation fails and an error is displayed.

  For example, the following command generates a map file named `myMapfile.txt`. If the specified map file already exists, it will be overwritten.

  ```
  wls:/mydomain/serverConfig>
  importMSASAppMetadata('/tmp/MSASartifacts.zip','/tmp/myMapfile.txt', true)
  ```

10.4 Migrating MSAS Application Metadata Between Application Environments

MSAS application metadata can be migrated through the different stages of the application development and deployment cycles, such as from development to production. Oracle recommends using the `exportMSASAppMetadata` and `importMSASAppMetadata` commands for policy migration, as described in "Exporting and Importing MSAS Application Metadata Using WLST" on page 10-2.

10.5 Replacing the MSAS Application Host and Port Values

To replace the MSAS application target host and port values, use the `migrateMSASAppHostports` command.

```
migrateMSASAppHostports(instanceName,applicationName,mapFileName,[generateMapFlag=
'false'])
```

Note the following:

- Use the `instanceName` argument to specify the name of the MSAS instance. In this case, a wildcard character is not accepted. The MSAS instance name is case-sensitive.
- Use the `applicationName` argument to specify the MSAS application whose referenced back-end service `host:port` information needs to be migrated or replaced. A wildcard '%' is allowed. If this argument is set to None, empty `''`, or `'%'`, then all applications in the specified MSAS instance are searched. The application name is not case-sensitive.
- Use the `mapFileName` argument to specify the location of a input map file that describes how to map source MSAS `host:port` values to a target `host:port`.

  For example, the following command migrates the `host:port` values for all applications that begin with `myApp` on the `MSAS-1234` instance according to the `myMapfile.txt` map file.

  ```
  wls:/mydomain/serverConfig> migrateMSASAppHostports('MSAS-1234', 'myApp%',
  '/tmp/myMapfile.txt')
  ```

- You can generate a new map file by setting the `generateMapFlag` argument to `true`. If the map file already exists, it will be overwritten; however, no migration takes place in the MSAS repository. If you specify a map file without setting the `generateMapFlag`, or setting it to `false`, and the map file does not exist, the operation fails and an error is displayed.

  For example, the following command collects all `host:port` values for the `myApp` application on an instance named `MSAS-1234`, and puts it into a newly-generated map file named `generatedMapfile.txt`. (Note that if a specified map file already exists, it will be overwritten).

  ```
  wls:/mydomain/serverConfig>  migrateMSASAppHostports
  ('MSAS-1234','myApp','/tmp/generatedMapfile.txt',true)
  ```