

IBM Security Access Manager for Mobile
Version 8.0.0.1

Configuration Guide



IBM Security Access Manager for Mobile
Version 8.0.0.1

Configuration Guide



Note

Before using this information and the product it supports, read the information in "Notices" on page 129.

Edition notice

Note: This edition applies to version 8.0.0.1 of IBM Security Access Manager for Mobile (product number 5725-L52) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2013.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures v

Tables vii

About this publication ix

Access to publications and terminology ix

Accessibility x

Technical training. x

Support information. x

Statement of Good Security Practices x

Chapter 1. Getting Started 1

Chapter 2. Activating the product and buying support. 3

Chapter 3. Managing application interfaces 5

Chapter 4. Managing the runtime component 7

Chapter 5. Managing user registries 9

Chapter 6. Runtime security services external authorization service 11

Updating the runtime security services EAS library file 11

Client certificate authentication considerations. 12

 Configuring runtime security services for client certificate authentication 13

Chapter 7. Using the isamcfg tool 15

Configuring a Web Gateway Appliance reverse proxy instance from the appliance. 15

Configuring a Web Gateway Appliance reverse proxy instance from an external machine 15

Configuring a WebSEAL instance 16

Configuring WebSEAL in a highly available environment 17

isamcfg reference 19

 isamcfg command line reference 19

 isamcfg Web Gateway Appliance configuration worksheet. 21

 isamcfg WebSEAL configuration worksheet. 25

Using a response file 28

Chapter 8. Support for compliance with NIST SP800-131a 31

Chapter 9. One-time passwords 35

One-time password configuration overview 35

One-time password delivery methods and providers 37

 Configuring an HOTP one-time password provider 38

 Configuring a TOTP one-time password provider 40

 Configuring a MAC one-time password provider 41

 Configuring an RSA one-time password provider 43

 Configuring one-time password delivery methods 46

 Managing mapping rules. 49

 One-time password delivery and user customization 55

Chapter 10. OAuth 2.0 support 67

OAuth 2.0 concepts. 67

OAuth 2.0 endpoints 68

OAuth 2.0 workflow 69

 Client authentication considerations at the OAuth 2.0 token endpoint 74

State management 74

Trusted clients management 75

Configuring API protection 76

 Creating API protection definition. 76

 Managing API protection definitions 79

 PIN policy. 80

 Registering an API protection client 81

 Managing registered API protection clients 82

 Managing policy attachments 83

 Uploading OAuth response files 86

Managing OAuth 2.0 mapping rules 86

 OAuth 2.0 mapping rule methods 87

OAuth 2.0 template pages 91

 OAuth 2.0 template page for consent to authorize 91

 OAuth 2.0 template page for errors 94

 OAuth 2.0 template page for response 94

 OAuth 2.0 template pages for trusted clients management 95

Error responses 97

User self-administration tasks for OAuth 98

 Managing OAuth 2.0 authorization grants 98

 REST services for OAuth 2.0 authorization grant management 99

Chapter 11. Modifying template files 101

Default template files. 102

Default template files language support 104

Chapter 12. Managing advanced configuration 107

Advanced configuration properties 108

Chapter 13. Deploying pending changes. 119

Chapter 14. Tuning runtime application parameters and tracing specifications 121

Chapter 15. Options for handling session failover events 125
Option 1: No handling of failover events 125

Option 2: The distributed session cache. 125

Chapter 16. Call Java code from within JavaScript rules 127

Notices 129

Index 133

Figures

1. WebSEAL client in an environment with multiple IBM Security Access Manager for Mobile servers 18
2. Template for allerror.html 59
3. Template for error_generating_otp.html 60
4. Template for error_get_delivery_options.html 61
5. Template for error_otp_delivery.html 62
6. Template for error_sts_invoke_failed.html 62
7. Template for error_could_not_validate_otp.html 63
8. Template page for sms_message.xml 64
9. Template for email_message.xml 64
10. OAuth 2.0 JavaScript sample code with state management 75
11. Template for user_consent.html 93
12. HTML template for user_error 94
13. Template for user_response.html 95
14. Template for clients_manager.html 97

Tables

| | | | |
|---|----|---|-----|
| 1. Runtime security services EAS access decisions | 11 | 8. Default template files in the ac/ directory | 102 |
| 2. Worksheet for Configuring Web Gateway Appliance from the appliance | 24 | 9. Default template files in the mga/ directory | 102 |
| 3. Worksheet for Configuring WebSEAL | 27 | 10. One-time password template files. | 103 |
| 4. One-time password template files | 55 | 11. Default files in the proper/ directory | 104 |
| 5. REST services instructions. | 65 | 12. OAuth template files | 104 |
| 6. OAuth 2.0 endpoint definitions and URLs | 68 | 13. Directory names and supported languages | 104 |
| 7. REST services instructions. | 99 | 14. Configuration data types. | 107 |
| | | 15. Filter by Category | 108 |

About this publication

The *IBM Security Access Manager for Mobile Configuration Guide* explains how to complete the initial configuration of IBM Security Access Manager for Mobile.

Access to publications and terminology

This section provides:

- A list of publications in the “IBM Security Access Manager for Mobile library.”
- Links to “Online publications.”
- A link to the “IBM Terminology website.”

IBM Security Access Manager for Mobile library

The following documents are available online in the IBM Security Access Manager for Mobile library:

- *IBM Security Access Manager for Mobile Configuration Guide*, SC27-6205-00
- *IBM Security Access Manager for Mobile Administration Guide*, SC27-6207-00
- *IBM Security Access Manager Appliance Administration Guide*, SC27-6206-00
- *IBM Security Access Manager for Mobile Auditing Guide*, SC27-6208-00
- *IBM Security Access Manager for Mobile Troubleshooting Guide*, GC27-6209-00
- *IBM Security Access Manager for Mobile Error Message Reference*, GC27-6210-00

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security IBM Security Access Manager for Mobile library

The product documentation site (http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.ammob.doc_8.0.0/welcome.html) displays the welcome page and navigation for the library.

IBM Security Systems Documentation Central

IBM Security Systems Documentation Central provides an alphabetical list of all IBM Security Systems product libraries and links to the online documentation for specific versions of each product.

IBM Publications Center

The IBM Publications Center site (<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>) offers customized search functions to help you find all the IBM publications you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at <http://www.ibm.com/software/globalization/terminology>.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. You can use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the IBM Accessibility website at <http://www.ibm.com/able/>.

Technical training

For technical training information, see the following IBM Education website at <http://www.ibm.com/software/tivoli/education>.

Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>.

IBM Security Access Manager for Mobile Troubleshooting Guide provides details about:

- What information to collect before contacting IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The **Community and Support** tab on the product information center can provide additional support resources.

Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

Chapter 1. Getting Started

Several configuration tasks are required to get started using IBM Security Access Manager for Mobile.

Complete each of the following tasks:

1. Activate the product.
2. Manage application interfaces.
3. Configure the runtime environment.
4. Manage user registries.
5. Update the runtime security services EAS library file.
6. Run the isamcfg tool.

Chapter 2. Activating the product and buying support

Activate the product after installation so you can use all available features. You can optionally import a support license file to receive updates to the appliance.

Before you begin

Obtain your activation key and support license:

- Download your activation key from your account on Passport Advantage at <https://www-112.ibm.com/software/howtobuy/softwareandservices>.
- Obtain your support license by following the instructions in the Welcome email that was sent by IBM.

Note: If you cannot locate your Welcome email, go to the IBM Security Systems License Key Center at <https://ibmss.flexnetoperations.com>. Review the FAQs to find out how to obtain support.

About this task

You can complete the following actions from the Support License and Product Activation panel:

- Import the activation key, which is required.
- Import the support license, which is optional. Import the license if you want to install service release updates.

The activation key is a permanent activation for the product. Activation keys have no expiration date.

Entitlement for X-Force updates for the database is provided automatically with the product. A third-party geolocation database is provided with sample data. You must purchase separately the full set of geolocation data.

You can review activation and support license information for your installed product packages, including specific product activations, service agreements, and expiration dates from this panel.

Procedure

1. Log in to the local management interface.
2. Click **Manage System Settings > Licensing and Activation**.
3. Perform one or more of the following actions:
 - Import the activation key and deploy the changes:
 - a. In the Licensing and Activation window, click **Import** under **Activated products**.
 - b. Browse to the activation key file that you downloaded from Passport Advantage.
 - c. Select the activation file.
 - d. Click **Open**.
 - e. Click **Save Configuration**.
 - f. Deploy the changes:

Note: You do not need to deploy changes immediately after you install the activation key. However, you must deploy changes before you can take a snapshot of the product.

- 1) In the undeployed change message, click **Click here to review the changes or apply them to the system**.
 - 2) Click **Deploy**.
- g. The activated product name and version are displayed in the Products table. To view the software license agreement, click: **View Service Agreement**.
- Optional: Import the product support license so that you can update the appliance:
 - a. In the Licensing and Activation window, under **Support license**, click **Select License**.
 - b. Browse to the support license file that you downloaded from IBM Security Systems License Key Center.
 - c. Select the license file.
 - d. Click **Save Configuration**.

Results

The menu in the local management interface refreshes to show the menu for the activated product.

If you imported a Support license, you can update the product with service releases. See "Installing updates" in the *IBM Security Access Manager for Mobile Appliance Administration Guide*.

Attention: Ensure that the activation is completed before attempting any other activities using the local management interface. For example, if you attempt to set up auditing or Secure Mobile Settings before activation is completed, an error occurs.

Chapter 3. Managing application interfaces

To manage application interfaces with the local management interface, use the Application Interfaces management page.

Procedure

1. From the top menu, select **Manage System Settings > Network Settings > Application Interfaces**. All current application interfaces are displayed in tabs. Each tab contains the current addresses and settings for a particular interface.
2. Select the tab of the interface that you want to work with. You can then add, edit, or delete an address on the corresponding interface tab.
 - **Add an address**
 - a. Click **New**.
 - b. In the Add Address page, provide details of the address to add.
 - Select the **Enabled** check box if you want this address to be enabled after creation.
 - Select **IPv4** or **IPv6** to indicate the type of address to add.
 - If **IPv4** is selected:
 - 1) Under **IPv4 Settings**, select either **Static** or **Auto** to indicate whether the IPv4 address is static or DHCP-assigned.

Note: Only one address per interface can be set to auto. If an existing address is already set to auto, then the **Auto** check box is disabled.
 - 2) *Optional:* If **Static** is selected in the previous step, you must enter the IPv4 address and subnet mask. If **Auto** is selected in the previous step, you can ignore the **Address** and **Subnet Mask** field.
 - If **IPv6** is selected, enter the IPv6 **Address** and **Prefix**.
 - Click **Save**.
 - **Modify an address**
 - *Method 1:*
 - a. Select the address to modify from the table.
 - b. Click **Edit**.
 - c. In the Edit Address page, modify as needed. See the “Add an address” section for descriptions of the fields.
 - d. Click **Save** to save your changes.
 - *Method 2:*
 - a. In the table, double-click the field to edit.
 - b. Make changes inline.

Note: Only some fields can be edited inline.
 - c. Click outside the editing field to save the changes.
 - **Delete an address**
 - a. Select the address to delete from the table.
 - b. Click **Delete**.
 - c. In the Delete Address page, click **Yes** to confirm the deletion.

- **Test connection to a server**
 - a. Click **Test**.
 - b. On the Ping Server page, enter the IP address or name of the server to test the connection with.
 - c. Click **Test**. A message is then displayed indicating whether the ping operation was successful.

Chapter 4. Managing the runtime component

To manage configuration files with the local management interface, use the Runtime Component management page.

About this task

When you first install the appliance, you must configure the runtime component. At any time after configuration, you can either edit the configuration settings, or unconfigure the runtime component.

Procedure

1. From the top menu, select **Secure Mobile Settings > Manage > Runtime Component**.
2. Select one of the following actions.

- **Configure**

- a. On the Main tab, select the **User Registry** type of LDAP.

Note: The runtime component does not communicate with the user registry, but you must select a registry type. It does not matter what user registry your system uses. Selection of the user registry type has no effect on the runtime component. Select LDAP in order to minimize the configuration steps.

- b. On the Policy Server tab, provide settings.
 - **Host name:** The name of the host that hosts the IBM® Security Access Manager policy server.
 - **Port:** The port over which communication with the IBM Security Access Manager policy server takes place.
 - **Management Domain:** The IBM Security Access Manager domain name.
- c. Provide settings on the LDAP tab. Enter a **Host name** and accept the default value for **Port**.

Note: The host name and port values are just placeholders. The runtime component does not use the values, but the configuration wizard requires that values must be set. If you selected Active Directory for the directory server, you must instead provide placeholder values for the fields on the **Active Directory SSL** and **Active Directory SSL** tabs. The values that you enter in these tab do not affect operation of the runtime component.

- d. Click **Finish** to save the settings.
- **Unconfigure** the remote policy server
 - a. Select the **Force** check box if you want the unconfigure operation to forcefully remove all of the configuration data. By default, this check box is not selected.

Note: Select the **Force** check box only if the unconfiguration fails repeatedly. Use this option only as a last resort.

- b. Click **Submit** to confirm operation.

- **Edit**

- a. Select the runtime configuration file of interest.
- b. Edit the configuration file and then click **Save** to save the changes. If you do not want to save the changes, click **Cancel**. If you want to revert to the previous version of the configuration file, click **Revert**.

Note: For the changes to take effect, they must be deployed.

Related information:

Deploy pending changes

Some configuration and administration changes require an extra deployment step.

Chapter 5. Managing user registries

The appliance runtime profile has a user registry associated. Use the User Registry management page to administer the users and group memberships. The user registry in discussion here is the one used by the runtime applications, not the one used by the management interface.

Procedure

1. From the top menu, select **Secure Mobile Settings > Manage > User Registry**. A list of all the current users in the registry is displayed. You can filter and reorder the list of users.
2. Perform one or more of the following actions as needed:

Create a user in the registry

- a. Click **New**.
- b. In the Create User window, enter the user name, password, and confirmation password for the new user.
- c. Click **OK**.

Delete a user from the registry

- a. Select the user to delete.
- b. Click **Delete**.
- c. In the window, click **Yes** to confirm the delete operation.

Change the password of a user in the registry

- a. Select the user for which you want to change password.
- b. Click **Set Password**.
- c. In the Set Password window, enter the password in the **New Password** and **Confirm Password** fields.
- d. Click **OK**.

Manage group memberships of a user

- a. Select the user of interest. The group memberships that are associated with this user are displayed under the **Group Membership** section.
- b. You can add the user to a group or delete the user from a group in the registry.

Add the user to a group

- 1) In the **Group Membership** section, click **Add**.
- 2) In the Add to Group window, select the group to add this user to.

Note: Only a single group can be selected.

- 3) Click **OK**.

Remove the user from a group

- 1) In the **Group Membership** section, select the group to remove the user from.
- 2) Click **Delete**.
- 3) In the window, click **Yes** to confirm the removal.

3. When you make changes to the user registry, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Chapter 6. Runtime security services external authorization service

The runtime security services external authorization service (EAS) provides the policy enforcement point function for context-based access.

You can configure the runtime security services EAS to include context-based access decisions as part of the standard authorization on WebSEAL requests. WebSEAL becomes the authorization enforcement point for access to resources that context-based access protects.

The runtime security services EAS constructs a request that it sends to the policy decision point (PDP). Based on the policy decision that is received from the PDP, the EAS takes one of the actions listed in the following table.

Table 1. Runtime security services EAS access decisions.

| Action | Description |
|-------------------------|--|
| Permit | Grants access to the protected resource. |
| Permit with obligations | Grants access to the protected resource, after the user successfully authenticates with a secondary challenge. |
| Deny | Denies access to the protected resource. |

The following steps set up the initial integration with IBM Security Access Manager for Mobile:

1. Update the RTSS EAS library file for WebSEAL.
2. Run the **isamcfg** tool to automatically update the WebSEAL configuration file and to complete other configuration setup.

Optionally, you can define custom attributes. See the topic "Custom attributes for the authorization service" in the *IBM Security Access Manager for Mobile Administration Guide*.

See the IBM Security Access Manager for Web information center for information about WebSEAL at http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.isam.doc_80/webseal.html.

Updating the runtime security services EAS library file

You might need to update your WebSEAL server with a newer runtime security services EAS library file. Without this update, the policy enforcement point functionality that IBM Security Access Manager for Mobile requires will not work properly.

Before you begin

Determine if you need to apply the update using the following list of supported WebSEAL versions:

IBM Security Access Manager for Web 7.0.0.1 or earlier hardware appliance

The update is already in place with the refresh to the runtime security services EAS library file. No action is necessary.

IBM Security Access Manager for Web 7.0.0.2 or earlier

Apply the update. Follow the steps below using the IBM Security Access Manager for Mobile 8.0 appliance local management interface.

IBM Tivoli Access Manager 6.1.1.7 or earlier

Apply the update. Follow the steps below using the IBM Security Access Manager for Mobile 8.0 appliance local management interface.

Procedure

1. Log in to the local management interface.
2. Click **Manage System Settings**.
3. Under **Secure Settings**, click **File Downloads**.
4. Locate the directory that contains the files you require. Click on the plus sign to expand the directories:
 - Expand mga.
 - Expand cba.
 - Expand EAS.
 - Expand 611 or 700, depending on the version of Access Manager you are using.
 - Select the appropriate operating system directory.
5. Select the library file, which is either `librtsseas.*` or `rtsseas.dll`.
6. Click **Export**.
7. Save the file to the computer where your browser is running.
8. Copy the library file to your WebSEAL server and replace the existing file. The file location is:

UNIX `/opt/pdweb/lib`

Windows

`INSTALL_DIR\PDWeb\bin`

where `INSTALL_DIR` is the installation location. Typically, this location is `C:\Program Files\Tivoli\PDWeb\bin`.

9. Restart your WebSEAL server instance:
 - a. `pdweb stop`
 - b. `pdweb start`

Client certificate authentication considerations

General considerations when choosing client certificate authentication.

Before choosing to use the client certificate authentication option provided in the `isamcfg` tool, you must:

- Generate a certificate that represents the user who will be authenticating from WebSEAL or the Web Reverse Proxy to Security Access Manager for Mobile for example, `easuser`
- Import that certificate into the WebSEAL or Web Reverse Proxy key database as a personal certificate
- Import the signer of this certificate as a trusted certificate in the Security Access Manager for Mobile keystore
- Set **Accept Client Certificates** to True on the Security Access Manager for Mobile appliance

- When answering the question 'Select the method for authentication between WebSEAL and the Security Access Manager for Mobile application interface' in the **isamcfg** tool select Certificate Authentication
- When prompted to enter the Security Access Manager for Mobile application interface SSL keyfile label enter the label of the certificate that represents the user who will be authenticating from WebSEAL or the Web Reverse Proxy to Security Access Manager for Mobile

For more information, see “Configuring runtime security services for client certificate authentication.”

Configuring runtime security services for client certificate authentication

Configure runtime security services for client certificate authentication used for authentication between WebSEAL and the Security Access Manager for Mobile appliance interface.

About this task

The provided steps are specific to Security Web Gateway Appliance version 7.0, but can be applied on the IBM Security Access Manager version 7.0 software product (WebSEAL).

Procedure

1. Create a client certificate for user **easusercert**.
 - a. In the local management interface, go to **Security Reverse Proxy Settings > Global Keys > SSL Certificates**.
 - b. Select the **pdsrv** certificate database.
 - c. Click **Manage > Edit SSL Certificate Database**.
 - d. Click **Personal Certificates**.
 - e. Click **New** to create a new personal certificate.
 - f. Provide the following information:
 - Certificate Label: easusercert
 - Certificate Distinguished Name: cn=easuser
 - Key Size: 2048
 - Expiration Time (in days): 365
 - g. Click **Save**.
2. Deploy pending changes. See Chapter 13, “Deploying pending changes,” on page 119.
3. Restart your reverse proxy instances.
4. Export the client certificate.
 - a. Select the **pdsrv** certificate database.
 - b. Click **Manage > Edit SSL Certificate Database**.
 - c. Click **Personal Certificates**.
 - d. Select the **easusercert** certificate you created.
 - e. Click **Manage > Export**.
 - f. Save the file.

5. Import the exported personal certificate as a signer certificate on the appliance. The signer of the client certificate needs to be trusted. The certificate is self-signed. Importing the **easusercert** as a signer certificate into the appliances allows that trust.
 - a. Click **Manage System Settings > Secure Settings > SSL Certificates**.
 - b. Select the **rt_profiles_keys** certificate database.
 - c. Click **Manage > Edit SSL Certificate Database**.
 - d. Click **Signer Certificates**.
 - e. Click **Manage > Import**.
 - f. Click **Browse**.
 - g. Browse to the directory that contains the file to be imported and select the file. Click **Open**.
 - h. Click **Import**. A message that indicates successful import is displayed.
6. Deploy pending changes. See Chapter 13, “Deploying pending changes,” on page 119.
7. Configure the appliance for client certificate authentication.
 - a. In the local management interface, go to **Secure Mobile Settings > Runtime Parameters > Runtime Tuning Parameters**.
 - b. Select **Accept Client Certificates**.
 - c. Click **Edit** and set the value as True.
8. Restart the runtime.

What to do next

Run the **isamcfg** tool and select **Certificate authentication** as the method of authentication between WebSEAL and the Security Access Manager for Mobile appliance interface. For more information, see “isamcfg Web Gateway Appliance configuration worksheet” on page 21.

Chapter 7. Using the `isamcfg` tool

Various features in IBM Security Access Manager for Mobile can be configured with the `isamcfg` tool.

The `isamcfg` tool helps automate some of the WebSEAL or Web Gateway Appliance configuration steps, including:

- Creates a junction that points to the IBM Security Access Manager for Mobile runtime endpoint.
- Creates an IBM Security Access Manager POP used by the IBM Security Access Manager for Mobile appliance to attach a policy.
- Configures the plug-in on WebSEAL or the Web Gateway Appliance that communicates to the IBM Security Access Manager for Mobile authorization server.
- Configures SSL, sets up key stores, trust stores and authentication configuration between IBM Security Access Manager for Mobile and WebSEAL or the Web Gateway Appliance.
- Configures a set of default obligation to URL mappings for the authentication service.
- Modifies WebSEAL or the Web Gateway Appliance authentication configuration to support the authentication service.

Configuring a Web Gateway Appliance reverse proxy instance from the appliance

Use the `isamcfg` tool to configure a Web Gateway Appliance reverse proxy instance from the appliance.

About this task

Run the following commands from the IBM Security Access Manager for Mobile command-line interface.

Procedure

1. Connect to the IBM Security Access Manager for Mobile appliance with SSH.
2. Enter the administrator user ID and password.
3. Navigate to `isam > mga > config`. The `isamcfg` tool starts.
4. Use the `isamcfg` to complete the configuration. For configuration details, see “`isamcfg` Web Gateway Appliance configuration worksheet” on page 21.

Results

When you complete the configuration, a summary screen displays indicating that the configuration is complete.

Configuring a Web Gateway Appliance reverse proxy instance from an external machine

Use the `isamcfg` tool to configure a Web Gateway Appliance reverse proxy instance from a remote machine.

Before you begin

Make sure that your Web Gateway Appliance server and Security Access Manager for Mobile servers are listening for connections on the appropriate management IP addresses and port numbers. To use the **isamcfg** tool, you must meet the following conditions:

- Obtain an IBM® JRE v6.0 Update 10 or later.
- At least one reverse proxy instance exists on the Web Gateway Appliance.
- Configure the `com.ibm.security.cmskeystore.CMSProvider` in the **java.security** file, which is in `$JAVA_HOME/lib/security`, of the IBM® JRE. The **isamcfg** tool uses the **ikeycmd** command to manipulate Key Data Base files. This requires the JRE to have the CMS provider configured in the **java.security** file.
- Ensure that the **ikeycmd** tool in the `$JAVA_HOME/bin` is on the system path.

Procedure

1. Download the **isamcfg.jar** from the IBM Security Access Manager for Mobile.
2. From the command line, type:

```
java -jar isamcfg -action config -cfgurl http://wga-appliance-host-url/
```
3. Use the **isamcfg** tool to complete the configuration. For configuration details, see “isamcfg Web Gateway Appliance configuration worksheet” on page 21.

Results

When you complete the configuration, a summary screen displays indicating that the configuration is complete.

Configuring a WebSEAL instance

Use the **isamcfg** tool to configure an IBM Security Access Manager WebSEAL as point of contact and policy enforcement point for IBM Security Access Manager for Mobile.

Before you begin

Make sure that your WebSEAL server is listening for connections on the appropriate IP addresses and port numbers. You can control the IP address and port number by using the WebSEAL configuration file. The IP address is controlled by the [server] network-interface configuration option, and the port numbers are controlled by the [server] https-port and [server] http-port options.

To use the **isamcfg** tool, you must:

- Obtain an IBM JRE, version 6.0, Update 10 or later that is supported by the version of PDJRte installed.
- Ensure that the Java Runtime used to launch the **isamcfg** tool has been configured into the Security Access Manager domain in full mode using the PDJRTE. An error is displayed if this condition is not met. For more information about using the PDJRTE, see the *IBM Developer Kit and Runtime Environment Java Technology Edition iKeyman User's Guide*.
- Ensure that the **isamcfg** tool is able to access the application interface for the mobile appliance.

For IBM Security Access Manager WebSEAL, version 7.0 or later, you must also meet the following conditions:

- Configure the `com.ibm.security.cmskeystore.CMSProvider` in the `java.security` file, which is in `$JAVA_HOME/lib/security`, of the IBM JRE. The `isamcfg` tool uses the `ikeycmd` command to manipulate Key Data Base files. This requires the JRE to have the CMS provider configured in the `java.security` file.
- Ensure that the `ikeycmd` tool in the `$JAVA_HOME/bin` is on the system path.

For Tivoli Access Manager for e-business WebSEAL versions 6.1.1 or prior, ensure that `gsk7ikm` tool is on the system path.

Run the tool on the same system where WebSEAL is located.

Procedure

1. Download the `isamcfg.jar` from the IBM Security Access Manager for Mobile.
2. On the WebSEAL machine, set up a `JAVA_HOME` environment variable for the JRE: For example:

```
export JAVA_HOME=/opt/ibm/java-x86_64-60/jre, or
export JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/jre
```
3. Add `$JAVA_HOME/bin` to the path `export PATH=$JAVA_HOME/bin:$PATH`.
4. From the command line, type:

```
java -jar isamcfg.jar -action config -cfgfile /path/to/webseald.conf
```
5. Use the `isamcfg` tool to complete the configuration. For configuration details, see “`isamcfg` WebSEAL configuration worksheet” on page 25.

Results

When you complete the configuration, a summary screen displays indicating that the configuration is complete.

Configuring WebSEAL in a highly available environment

When you are working in an environment with multiple IBM Security Access Manager for Mobile servers, you can configure WebSEAL for failover and high availability.

About this task

You can configure the WebSEAL junction and Runtime Security Services External Authorization Service (RTSS EAS) to take advantage of the IBM Security Access Manager for Mobile high availability.

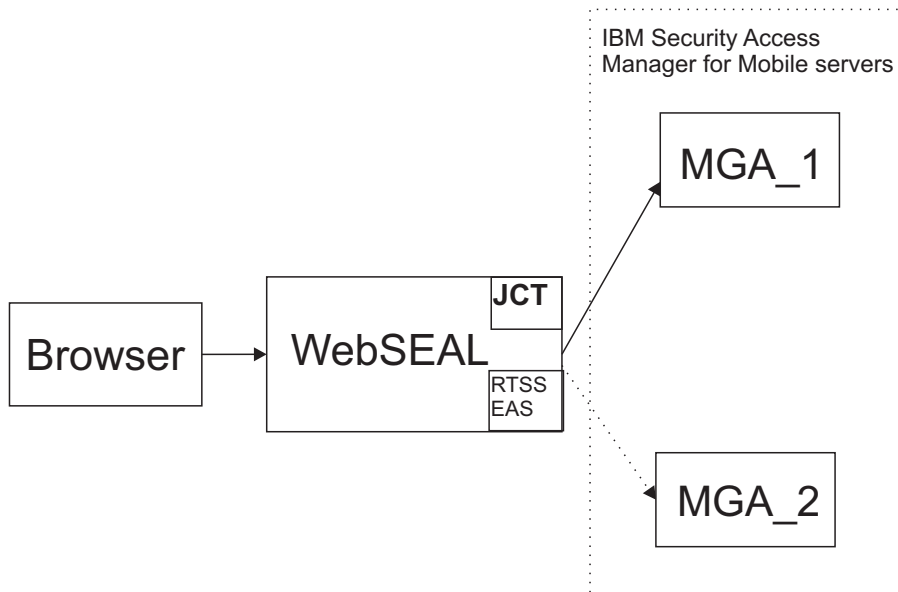


Figure 1. WebSEAL client in an environment with multiple IBM Security Access Manager for Mobile servers

The **isamcfg** tool is provided with IBM Security Access Manager for Mobile. You must run this configuration tool to configure each WebSEAL instance for use with the IBM Security Access Manager for Mobile appliance. This tool sets up a single junction server and configures the RTSS EAS to point to a single appliance.

If you have more than one IBM Security Access Manager for Mobile appliance, you need to manually configure the additional servers.

Procedure

For each additional IBM Security Access Manager for Mobile appliance, you must complete the following configuration:

1. Update the WebSEAL configuration file (for example, `webseald-default.conf`) to add the additional servers to the RTSS EAS configuration.

Include a server entry in the `[rtss-cluster:<cluster>]` stanza for each appliance. For example:

```
[rtss-cluster:cluster1]
server = 9,https://9.48.167.40:443/rtss/authz/services/AuthzService
server = 9,https://9.48.167.117:443/rtss/authz/services/AuthzService
```

Note:

- The first parameter in each entry is the priority of the server in the cluster. Set the priority of your servers as appropriate to your environment. Using a priority of 9 for all servers evenly distributes the load and switches between the available appliances.
 - The second parameter is a well-formed Uniform Resource Locator (URL) for the runtime security services on the appliance. Use the IP address of the application interface on the IBM Security Access Manager for Mobile appliance.
2. Use the **pdadmin** utility to add extra servers to the junction. For example:

```
pdadmin sec_master> server task default-webseald-test.ibm.com add -h
9.48.167.40 -p 443 /mga
pdadmin sec_master> server task default-webseald-test.ibm.com add -h
9.48.167.117 -p 443 /mga
```

Note:

- You must replace all example values in these commands with values that are appropriate to your environment.
- The first parameter in this server task command is the fully qualified name of the WebSEAL server. For example, `default-webseald-test.ibm.com`.
- The `-h` option specifies the IBM Security Access Manager for Mobile appliance that you want to add to the junction. Use the IP address of the application interface on the target appliance.
- The `isamcfg` tool creates an SSL junction by default. Therefore, when you are adding servers to this junction, use the SSL port number 443.
- By default, the `isamcfg` tool creates a junction that is called `/mga`. This default value is used in the example commands.

You must also complete the following general configuration in the environment:

3. For secure communication between WebSEAL and the appliance, ensure that trusted certificates are used. WebSEAL must trust the certificates that are presented by IBM Security Access Manager for Mobile. To establish this trust, you can use a common certificate authority (CA) that is trusted in your environment or you can configure WebSEAL to trust each individual certificate. Similarly, for client certificate authentication, IBM Security Access Manager for Mobile must trust the certificates that are presented by WebSEAL.

4. To configure failover between junctioned servers, set the `use-new-stateful-on-error` stanza entry to `yes` for the stateful junction to the appliance. That is, update the `use-new-stateful-on-error` entry in the `[junction:/mga]` stanza in the WebSEAL configuration file. Where `/mga` is the name of the junction. The `isamcfg` tool creates a junction that is called `/mga` by default, but this name is configurable.

If a stateful junction becomes unavailable when this value is set to `yes`, WebSEAL fails over to a different server. For example, if the stateful junction to MGA_1 in Figure 1 on page 18 becomes unavailable, WebSEAL fails over to MGA_2.

isamcfg reference

Use the `isamcfg` tool to configure IBM Security Access Manager for Mobile WebSEAL or Web Gateway Appliance as point of contact and policy enforcement point.

isamcfg command line reference

Use the command line options described in this section to configure and unconfigure WebSEAL and Web Gateway Appliance servers.

Syntax

```
java -jar isamcfg.jar -action mode options
```

Description

The IBM Security Access Manager configuration tool has two modes of operation:

- **config**
- **unconfig**

Each mode uses different command line options.

Options

-action config *options*

This command configures a WebSEAL or Web Gateway Appliance server. This mode uses different command line options:

-cfgfile *file*

Specifies which WebSEAL configuration file to use. This option is required when configuring a WebSEAL server.

-cfgurl *URL*

Specifies the Web Gateway Appliance configuration URL to use. This option is required if configuring a Web Gateway Appliance

-rspfile *file*

Specifies the response file for a configuration that is not interactive.
Default value: Interactive configuration.

-record

Generates response file without making changes to WebSEAL or Web Gateway Appliance configuration.

-action unconfig *options*

This command unconfigures a WebSEAL or Web Gateway Appliance server. This mode uses different command line options:

-cfgfile *file*

Specifies which WebSEAL configuration file to use. This option is required when unconfiguring a WebSEAL server.

-cfgurl *URL*

Specifies the Web Gateway Appliance configuration URL to use. This option is required when unconfiguring a Web Gateway Appliance

-rspfile *file*

Specifies the response file for a configuration that is not interactive.
Default value: Interactive configuration.

-record

Generates response file without making changes to WebSEAL or Web Gateway Appliance configuration.

Example

```
java -jar isamcfg.jar -action -config -cfgfile webseald.conf
```

The log files for the **isamcfg** tool are written to the system temporary directory. The system temporary file directory is specified by the system property **java.io.tmpdir**.

isamcfg Web Gateway Appliance configuration worksheet

Use the **isamcfg** tool to configure a Web Gateway Appliance server from a command line.

This worksheet describes the prompts. Use this worksheet to plan your properties, and refer to it when running the tool.

Select/deselect the capabilities you would like to configure by typing its number.

- Context-based Authorization

Configure this capability if your environment requires the use of behavioral and contextual data analytics to calculate the risk of a transaction.

By default, the tool selects context-based authorization, authentication service, and API protection. You must deselect the capability that you do not want to configure by selecting its corresponding number. For example, press 1 to clear the context-based authorization box.

You can choose to configure context-based authorization and authentication service at the same time.

- Authentication service

Configure this capability if your environment requires the use of a step-up authentication type of authentication.

By default, the tool selects context-based authorization, authentication service, and API protection. You must deselect the capability that you do not want to configure by selecting its corresponding number. For example, press 2 to clear the authentication service box.

You can choose to configure context-based authorization and authentication service at the same time.

- API Protection

Configure this capability if your environment requires the use of an OAuth authentication type to protect your Application Protocol Interface (API).

By default, the tool selects context-based authorization, authentication service, and API protection. You must deselect the capability that you do not want to configure by selecting its corresponding number. For example, press 3 to clear the API protection box.

You can choose to configure context-based authorization, authentication service, and API protection at the same time.

Security Access Manager for Mobile Local Management Interface hostname

Enter the Local Management Interface hostname or IP address.

Security Access Manager for Mobile Local Management Interface port

Specify the port number of the Local Management Interface. The tool displays a port number.

Example value: 443

Press Enter to use the displayed port or enter your preferred port.

Security Access Manager for Mobile Appliance administrator user ID

Press Enter to use the displayed user ID or enter your preferred user ID.

Security Access Manager for Mobile Appliance administrator password

Enter the corresponding administrator password.

SSL certificate data valid (y/n)

Press **y** to validate that the displayed SSL certificate values are valid otherwise, press **n**.

Web Gateway Appliance Local Management Interface hostname

Enter the Web Gateway Appliance Local Management Interface hostname or IP address. The tool might display a value. Press Enter to use the displayed value or enter your preferred hostname or IP address.

Web Gateway Appliance Local Management Interface port

Specify the port number of the Local Management Interface port. The tool displays a port number.

Example value: 443

Press Enter to use the port or enter your preferred port.

Web Gateway Appliance administrator user ID

Press Enter to use the user ID or enter your preferred user ID.

Web Gateway Appliance administrator password

Enter the corresponding Web Gateway Appliance administrator password.

SSL certificate data valid (y/n)

Press **y** to validated that the displayed SSL certificate values are valid otherwise, press **n**.

Instance to configure

The tool displays the available instances you can configure in a list. Select the instance you would like to configure.

Security Access Manager for Mobile Appliance administrator user ID

Press Enter to use the displayed user ID or enter your preferred user ID.

Security Access Manager for Mobile Appliance administrator password

Enter the corresponding administrator password.

Security Access Manager for Mobile application interface hostname

Enter the hostname or IP address of the Security Access Manager for Mobile application interface.

Example value: 172.16.229.10

Security Access Manager for Mobile application interface port

Specify the port number of the Security Access Manager for Mobile application interface.

Example value: 443

Select the method for authentication between WebSEAL and the Security Access Manager for Mobile application interface

- Certificate authentication
Use a certificate to authenticate between WebSEAL and the Security Access Manager for Mobile application interface.
- User ID and password authentication
Use credentials to authenticate between WebSEAL and the Security Access Manager for Mobile application interface.

Security Access Manager for Mobile application interface user ID:

Press Enter to use the displayed user ID or enter your preferred user ID.

Security Access Manager for Mobile application interface password:

Enter the corresponding Security Access Manager for Mobile application interface password.

SSL certificate data valid (y/n):

Press **y** to validated that the displayed SSL certificate values are valid otherwise, press **n**.

Automatically add CA certificate to the key database (y/n)

Press **y** if you want to automatically add the CA certificate to the key database, otherwise press **n**.

Note: Web Reverse Proxy instance restarts if **y** is selected.

The following files are available on the Web Gateway Appliance.

Choose one for the **400 Bad Request** response page.

Note: If you are not running the isamcfg tool on the appliance, you can choose **Cancel** to upload a local file. If you are running the isamcfg tool on the appliance, you must upload your custom response file to the Web Gateway Appliance first before you run the isamcfg tool so that the file is displayed as an option. For more information, see "Uploading OAuth response files" on page 86.

The following files are available on the Web Gateway Appliance.

Choose one for the **401 Unauthorized** response page.

Note: If you are not running the isamcfg tool on the appliance, you can choose **Cancel** to upload a local file. If you are running the isamcfg tool on the appliance, you must upload your custom response file to the Web Gateway Appliance first before you run the isamcfg tool so that the file is displayed as an option. For more information, see "Uploading OAuth response files" on page 86.

The following files are available on the Web Gateway Appliance.

Choose one for the **502 Bad Gateway** response page.

Note: If you are not running the isamcfg tool on the appliance, you can choose **Cancel** to upload a local file. If you are running the isamcfg tool on the appliance, you must upload your custom response file to the Web Gateway Appliance first before you run the isamcfg tool so that the file is displayed as an option. For more information, see "Uploading OAuth response files" on page 86.

The junction mga contains endpoints that require Authorization HTTP header to be forwarded to the backend server. Do you want to enable this feature? [y|n]?

Press **y** to allow endpoints that require Authorization HTTP header to be forwarded to the backend server. Otherwise, press **n**.

Table 2. Worksheet for Configuring Web Gateway Appliance from the appliance

| Prompt | Your value |
|--|------------|
| Select/deselect the capabilities you would like to configure by typing its number. Press enter to continue: <input type="checkbox"/> 1. Context-based Authorization <input type="checkbox"/> 2. Authentication service <input type="checkbox"/> 3. API Protection | |
| Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel | |
| Security Access Manager for Mobile Local Management Interface hostname: | |
| Security Access Manager for Mobile Local Management Interface port: | |
| Security Access Manager for Mobile Appliance administrator user ID: Security Access Manager for Mobile Appliance administrator password: | |
| SSL certificate data valid (y/n): | |
| Web Gateway Appliance Local Management Interface hostname: | |
| Web Gateway Appliance Local Management Interface port: | |
| Web Gateway Appliance administrator user ID: | |
| Web Gateway Appliance administrator password: | |
| SSL certificate data valid (y/n): | |
| Instance to configure: Enter your choice: | |
| Security Access Manager administrator user ID: Security Access Manager administrator password: | |
| Security Access Manager for Mobile application interface hostname: | |
| Security Access Manager for Mobile application interface port: | |
| Select the method for authentication between WebSEAL and the Security Access Manager for Mobile runtime: <input type="checkbox"/> 1. Certificate authentication <input type="checkbox"/> 2. User ID/password authentication Enter you choice: | |
| Security Access Manager for Mobile application interface user ID: Security Access Manager for Mobile application interface password: | |
| SSL certificate data valid (y/n): | |
| Automatically add CA certificate to they key database (y/n): | |
| The following files are available on the Web Gateway Appliance. Choose one file for the 400 Bad Request response page | |
| The following files are available on the Web Gateway Appliance. Choose one file for the 401 Unauthorized response page | |
| The following files are available on the Web Gateway Appliance. Choose one file for the 502 Bad Gateway response page | |
| The junction /mga contains endpoints that require Authorization HTTP header to be forwarded to the backend server. (y n): | |

Note: The tool restarts Web Reverse Proxy without warning after implementing the configuration.

isamcfg WebSEAL configuration worksheet

Use the **isamcfg** tool to configure WebSEAL server from a command line.

This worksheet describes the prompts. Use this worksheet to plan your properties, and refer to it when running the tool.

Select/deselect the capabilities you would like to configure by typing its number.

- **Context-based Authorization**

Configure this capability if your environment requires the use of behavioral and contextual data analytics to calculate the risk of a transaction.

By default, the tool selects context-based authorization, authentication service, and API protection. You must deselect the capability that you do not want to configure by selecting its corresponding number. For example, press 1 to clear the context-based authorization box.

You can choose to configure context-based authorization and authentication service at the same time.

- **Authentication service**

Configure this capability if your environment requires the use of a step-up authentication type of authentication.

By default, the tool selects context-based authorization, authentication service, and API protection. You must deselect the capability that you do not want to configure by selecting its corresponding number. For example, press 2 to clear the authentication service box.

You can choose to configure context-based authorization and authentication service at the same time.

- **API Protection**

Configure this capability if your environment requires the use of an OAuth authentication type to protect your Application Protocol Interface (API).

By default, the tool selects context-based authorization, authentication service, and API protection. You must deselect the capability that you do not want to configure by selecting its corresponding number. For example, press 3 to clear the API protection box.

You can choose to configure context-based authorization, authentication service, and API protection at the same time.

Security Access Manager for Mobile Local Management Interface hostname

Enter the Local Management Interface hostname or IP address.

Security Access Manager for Mobile Local Management Interface port

Specify the port number of the Local Management Interface. The tool displays a port number.

Example value: 443

Press Enter to use the displayed port or enter your preferred port.

Security Access Manager for Mobile Appliance administrator user ID

Press Enter to use the displayed user ID or enter your preferred user ID.

Security Access Manager for Mobile Appliance administrator password

Enter the corresponding administrator password.

Security Access Manager Domain name

Enter the Security Access Manager domain name. Press Enter to use the default domain name or enter your preferred domain name.

Security Access Manager administrator user ID

Enter a valid Security Access Manager administrator user ID. Press Enter to use the user ID or enter your preferred user ID.

Security Access Manager administrator password

Enter the corresponding Security Access Manager administrator password.

Security Access Manager for Mobile application interface hostname

Enter the hostname or IP address of the Security Access Manager for Mobile application interface.

Example value: 172.16.229.10

Security Access Manager for Mobile application interface port

Specify the port number of the Security Access Manager for Mobile application interface.

Example value: 443

Security Access Manager for Mobile application interface SSL key file

Specify the path to a keystore that contains the SSL keys required to connect to the Security Access Manager for Mobile application interface or press Enter to use the default key file.

Security Access Manager for Mobile application interface SSL stash file

Specify the path to a stash file that contains the password to the Security Access Manager for Mobile application interface SSL keyfile or press Enter to use the default stash file.

Select the method for authentication between WebSEAL and the Security Access Manager for Mobile application interface

- Certificate authentication
Use a certificate to authenticate between WebSEAL and the Security Access Manager for Mobile application interface.

Note: On Windows operating systems, you must use certificate authentication for WebSEAL from IBM Security Access Manager for Web 7.0.0.2.

- User ID and password authentication
Use credentials to authenticate between WebSEAL and the Security Access Manager for Mobile application interface.

Security Access Manager for Mobile application interface SSL key file label

Specify the key label of the certificate to present to the Security Access Manager for Mobile runtime.

SSL certificate data valid (y/n)

Press **y** to validated that the displayed SSL certificate values are valid otherwise, press **n**.

Runtime security service external authorization service library

By default, the tool displays the available library. Press Enter to use the available library or enter your preferred library.

The 400 Bad Request response page:

The default location of the response page is displayed.

The 401 Unauthorized response page:

The default location of the response page is displayed.

The 502 Bad Gateway response page:

The default location of the response page is displayed.

The junction mga contains endpoints that require Authorization HTTP header to be forwarded to the backend server. Do you want to enable this feature? [y|n]?

Press y to allow endpoints that require Authorization HTTP header to be forwarded to the backend server. Otherwise, press n.

Note: The tool restarts WebSEAL without warning after the configuration is implemented.

Table 3. Worksheet for Configuring WebSEAL

| Prompt | Your value |
|--|------------|
| Select/deselect the capabilities you would like to configure by typing its number. Press enter to continue: <input type="checkbox"/> 1. Context-based Authorization <input type="checkbox"/> 2. Authentication service <input type="checkbox"/> 3. API Protection | |
| Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel | |
| Security Access Manager for Mobile Local Management Interface hostname: | |
| Security Access Manager for Mobile Local Management Interface port: | |
| Security Access Manager for Mobile Appliance administrator user ID: Security Access Manager for Mobile Appliance administrator password: | |
| SSL certificate data valid (y/n): | |
| Security Access Manager Domain name: | |
| Security Access Manager administrator user ID: Security Access Manager administrator password: | |
| Security Access Manager for Mobile application interface port: | |
| Security Access Manager for Mobile application interface SSL key file: | |
| Security Access Manager for Mobile application interface SSL stash file: | |
| Select the method for authentication between WebSEAL and the Security Access Manager for Mobile application interface: <input type="checkbox"/> 1. Certificate authentication <input type="checkbox"/> 2. User ID/password authentication Enter you choice: | |
| Security Access Manager for Mobile application interface SSL key file label | |
| SSL certificate data valid (y/n): | |
| Runtime security service external authorization service library | |
| The 400 Bad Request response page | |
| The 401 Unauthorized response page | |
| The 502 Bad Gateway response page | |

Table 3. Worksheet for Configuring WebSEAL (continued)

| Prompt | Your value |
|---|------------|
| The junction /mga contains endpoints that require Authorization HTTP header to be forwarded to the backend server. (y n): | |

Using a response file

Create and use a response file with the **isamcfg** tool to configure WebSEAL or Web Gateway Appliance.

A response file records the actions to be taken by the **isamcfg** tool.

Use the **-record** command to create a response file without changing the WebSEAL or Web Gateway Appliance configuration. For example:

```
/usr/lib/jvm/jre-1.7.0-ibm.x86_64/bin/java -jar
/opt/IBM/FIM/tools/isamcfg/isamcfg.jar -action config
-cfgurl https://1.1.1.1/ -record
```

You can then use the response file to run the **isamcfg** tool non-interactively. Use the **-rspfile** command to run the **isamcfg** tool with a response file. For example:

```
/usr/lib/jvm/jre-1.7.0-ibm.x86_64/bin/java -jar
/opt/IBM/FIM/tools/isamcfg/isamcfg.jar -action config
-cfgurl https://1.1.1.1/ -rspfile /tmp/isamcfg-mobile-mobile.properties
```

The contents of a response varies depending on your configuration, for example:

```
#Fri Sep 06 12:45:28 EST 2013
webseal.addcacert=y
tam.admin=sec_master
wga.pass=
pop.replace=pop.reuse
mga.admin.ssl.shalfingerprint=CD\1C\F0\D9\A6\3A\7A\11\
:16\CC\18\CB\56\02\08\E2\53\99\83\3B
mga.runtime.auth.mode=mga.runtime.auth.ba
mga.runtime.ssl.md5fingerprint=E4\65\16\A9\D8\B2\97\
:3C\F6\13\19\77\25\8B\B0\0A
mga.admin.ssl.subjectdn=CN\=amapp800
wga.ssl.md5fingerprint=B2\9F\87\8A\D1\49\D9\A1\BA\
:03\4B\41\E9\DF\44\C7
rtss.password=
wga.ssl.shalfingerprint=CD\1C\F0\D9\A6\3A\7A\11\16\
:CC\18\CB\56\02\08\E2\53\99\83\3B
wga.instance=mobile
mga.admin.user=admin
wga.port=443
isam.mode=context_based_authorization,
authentication_service,
mga.runtime.port=443
mga.admin.pass=
jct.replace=reuse
wga.host=1.1.1.1
mga.runtime.host=1.1.1.1
mga.admin.ssl.md5fingerprint=B2\9F\87\8A\D1\49\D9\
:A1\BA\03\4B\41\E9\DF\44\C7
wga.ssl.subjectdn=CN\=amapp800
mga.runtime.ssl.subjectdn=CN\=isam, O\=ibm, C\=us
mga.admin.ssl.issuerdn=CN\=amapp800
rtss.user=admin
mga.runtime.ssl.shalfingerprint=F9\38\5A\53\0A\DA\1A\
:FF\67\46\C9\58\3F\F1\2B\00\B0\6C\83\32
mga.admin.port=443
tam.password=
```



```
wga.ssl.issuerdn=CN\=amapp800  
mga.runtime.ssl.issuerdn=CN\=isam, O\=ibm, C\=us  
wga.user=admin  
mga.admin.host=1.1.1.1
```

Chapter 8. Support for compliance with NIST SP800-131a

IBM Security Access Manager for Mobile 8.0 supports the requirements that are defined by the National Institute of Standards and Technology (NIST) Special Publications 800-131a.

SP 800-131a strengthens security by defining stronger cryptographic keys and more robust algorithms. The standard defines a period to allow customers time to make the transition to the new requirements. The transition period closes at the end of 2013. See the NIST publication *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths* for the new standards that are defined by Special Publication 800-131, and details about allowed protocols, cipher suites, and key strength.

You can run IBM Security Access Manager for Mobile 8.0 in either of the two modes that are supported by NIST SP800-131a:

- Transition mode
- Strict mode

When configured in transition mode, server components support the transition mode Transport Layer Security (TLS) protocols, which include TLS 1.0 and TLS 1.1. Client components, such as the HTTPS client that performs one-time password (OTP) delivery and the syslog auditing client, support TLS 1.2 only.

When configured in strict mode, both the server components and the client components of IBM Security Access Manager for Mobile support TLS 1.2 only.

To deploy in transition mode, you need to select only the mode during initial configuration of the appliance. To run in strict mode, you must also set an extra configuration option.

If your deployment uses client certificate authentication, and you want to use strict mode, you must complete more configuration steps for the point of contact server. The point of contact server can be either IBM Security Access Manager WebSEAL or IBM Security Web Gateway Appliance 7.0.

Transition mode

When you install the appliance, select the option to enable FIPS 140-2 mode. This selection turns on compliance for NIST SP800-131a.

When enabled, NIST SP800-131a compliance is run in transition mode. You do not have to complete any further configuration steps in order to run in transition mode.

Note:

- Enable FIPS 140-2 mode only if you must comply with the NIST SP800-131a requirements. There is no advantage to enabling FIPS 140-2 mode if your installation does not require this compliance.

Important: The setting of the FIPS 140-2 Mode option is permanent and cannot be turned off after it is enabled. To disable the option, you must reinstall the appliance.

- If you enable FIPS 140-2 mode, the appliance is automatically restarted before it continues with the rest of the setup.
- FIPS Limitation: For IBM Security Access Manager for Mobile, the FIPS 140-2 mode option in the appliance setup wizard does not turn on compliance for FIPS 140-2. It turns on compliance for NIST SP800-131a only.

Strict mode

Overview of configuration tasks:

1. Enable FIPS 140-2 mode during appliance configuration.
2. Set a tuning parameter to enable strict mode.
3. (Optional) If your deployment uses client certificate authentication, configure TLS v1.2.

Instructions:

1. Install the appliance and choose to enable FIPS 140-2 mode. This selection turns on compliance for NIST SP800-131a.
2. Use the appliance local management interface (LMI) to modify the advanced tuning parameter `nist.sp800-131a.strict`. This parameter is set by default to false. Complete the following steps:
 - a. Verify that your browser supports TLS 1.2.
CAUTION:
Strict mode requires the use of TLS 1.2. Some browsers support TLS 1.2 but have the support disabled by default. If you set the value of the `nist.sp800-131a.strict` parameter to true, and your browser is not configured to support TLS 1.2, you lose access to the appliance LMI.
 - b. On the LMI, select **Manage System Settings > System Settings > Advanced Tuning Parameters**.
 - c. Select `nist.sp800-131a.strict`. Select **Edit**. Change the value to true.
3. Determine whether your deployment uses basic authentication or client certificate authentication, for communication between IBM Security Access Manager for Mobile and the point of contact server.
 - If you use basic authentication, the configuration is complete.
 - If you use client certificate authentication, continue with the next section.

Client certificate configuration for strict mode

If you use client certificate authentication on the point of contact server, you must configure it to be in compliance with NIST SP800-131a strict mode.

To comply with strict mode, configure the point of contact server to use TLS v1.2 for client certificate authentication.

You must create a self-signed certificate, and configure the point of contact server to use TLS v1.2 with the Runtime Security Services External Authorization Service (EAS). Complete each of the following tasks:

1. Create a self-signed certificate.
 - Review the topic “Client certificate authentication considerations” on page 12. Select one of the following actions, as fits your deployment:

- If your deployment uses the IBM Security Web Gateway Appliance (Web Reverse Proxy), follow the instructions in "Configuring runtime security services for client certificate authentication" on page 13. In Step 1 "Create a client certificate for user easusercert", specify:

Signature Algorithm: SHA2withRSA

- If your deployment uses WebSEAL:

Manually create a self-signed certificate. To specify a NIST-compliant algorithm, use an external utility such as **gsk7ikm**. Open the `pd.srv` certificate database, and create a self-signed certificate with these credentials:

```
Certificate Label: easusercert
Certificate Distinguished Name: cn=easuser
Key Size: 2048
Expiration Time (in days): 365
Signature Algorithm: SHA2withRSA
```

Note:

- The user `cn=easuser` is the built-in user, but any user with sufficient permissions (as created by the IBM Security Access Manager for Mobile administrator) can be used instead.
- It is not mandatory that WebSEAL has FIPS 140-2 mode configured in order to communicate with the IBM Security Access Manager for Mobile server. However, to comply with NIST SP800-131a strict mode, client certificate authentication between WebSEAL and the server must be over TLS v1.2.
- See the *IBM Security Access Manager for Web Version 7.0 WebSEAL Administration Guide* for complete information on configuring client certificate authentication.

2. Configure the point of contact server to use TLS v1.2 with the Runtime Security Services External Authorization Service (EAS)

The point of contact server uses the EAS to process authorization requests. The default EAS setting for communication specifies Secure Sockets Layer (SSL) v2, which is not supported by the IBM Security Access Manager for Mobile appliance when it operates in NIST SP800-131a strict mode. If you do not adjust the configuration setting for the EAS, the authorization request (and the regular ping call) does not succeed

Select the action that fits your deployment:

- If you deploy your point of contact server on the same computer as the appliance:
 - a. In the IBM Security Access Manager for Mobile appliance local management interface, select **Reverse Proxy Settings > your_instance_name > Manage > Configuration > Edit** to open the configuration file. Add the following parameter to the existing stanza:


```
[rtss-cluster:cluster1]
gsk-attr-name = enum:438:1
```
 - b. Click **Save**. Deploy the changes. Restart the instance.
- If you deploy your point of contact server on a different computer from the appliance:
 - a. Open the WebSEAL instance configuration file for editing. For example: `/opt/pdweb/etc/webseald-appliance-default.conf`.
 - b. Add the following parameter to the existing stanza:


```
[rtss-cluster:cluster1]
gsk-attr-name = enum:438:1
```

Chapter 9. One-time passwords

The IBM Security Access Manager for Mobile appliance can be configured to perform one-time password as an authentication factor in an authentication scenario.

The IBM Security Access Manager for Mobile appliance provides strong authentication mechanisms for an IBM Security Access Manager for Web point of contact interface.

The point of contact server is a proxy or application server that interacts with a user, does the authentication, and manages sessions. In a typical deployment, the point of contact is at the edge of a protected network behind a firewall, such as in a DMZ.

The authentication methods available in a deployment are typically determined by the point of contact technology that is used in the environment. Points of contact technologies usually provide simple authentication such as the use of a user name and password.

A step-up authentication is a type of authentication where users who attempt to access sensitive resources are required to provide a specific type of credential. They might be challenged to authenticate and provide an extra set of credentials to prove that they are allowed to access sensitive resources. The one-time password authentication can be used where increased security is required.

A multi-factor authentication is a type of authentication where users are required to provide more than one type of credential to access a protected resource. A one-time password is a unique password that validates a login session. A one-time password cannot be reused. These restrictions make it less vulnerable to replay attacks and more secure than static passwords.

The one-time password authentication capability in the IBM Security Access Manager for Mobile appliance provides the following features:

- One-time password generation and validation with support for various implementations out of the box.
- One-time password delivery with email and short message service (SMS) implementation.
- Time-based, counter-based, and RSA one-time password generation and validation that requires no delivery mechanism.

The one-time password authentication flow consists of allowing multi-factor and step-up authentication operations that rely on the one time password technologies supported by the IBM Security Access Manager for Mobile appliance. The one-time password authentication is used to extend the authentication capabilities of existing point of contact technologies.

One-time password configuration overview

Most of the one-time password authentication is pre-configured on the appliance. In the majority of scenarios, this configuration is adequate. However, some scenarios require customization to meet your requirements.

You can configure the following components to customize the one-time password functionality:

- Configure the point of contact settings.
- Customize the one-time password settings.
- Customize the template pages and one-time password rules to suit your requirements.

Point of contact settings

You can configure the point of contact in the administration console Advanced Configuration panel. For more information, see Chapter 12, “Managing advanced configuration,” on page 107.

The point of contact configuration allows for multiple authentication methods to be configured. Each authentication method is represented in the configuration by an authentication callback. The execution of an authentication event consists of invoking the list of configured authentication callbacks. Each configured callback is assigned an authentication level. The configured authentication level represents the level of assurance that each authentication callback provides. The required authentication level is determined by the configured authentication policy. If an authenticated session exists when the authentication event happens, the required authentication level determines if a particular token provided for the authenticated session is satisfactory. The authentication service can be configured to allow reauthentication. If enabled, the authentication service invokes all the configured authentication callbacks regardless of the pre-existing authentication session.

The required authentication level determines the level of authentication that is required of a user to be able to access a protected resource. Authentication levels are represented by an integer number. Each authentication callback is assigned an authentication level.

During an authentication event, the configured authentication callbacks are used. The required authentication policy is enforced. To evaluate if an authenticated session is satisfactory based on the policy, the appliance point of contact retrieves the authentication level of the credential. If no valid or satisfactory credential exists, the callbacks are started until a satisfactory level is achieved. You can modify the authentication level that is configured for each configured authentication callback.

The type of authentication determines the authentication that is required for a user to be able access a protected resource. There are two supported types of authentication:

Hierarchical authentication type (step-up)

Executes the authentication callback with an assigned authentication level that is equal to or higher than the required level. It is executed until a satisfactory authentication is achieved.

Complementary authentication type (multi-factor)

Executes all the authentication callbacks that are configured until a satisfactory authentication is achieved.

Authentication policy

User access to a resource is also determined by the *authentication policy*. An authentication policy:

- Applies a set of rules to the authentication process and to the verification of authentication data.
- Determines enforcement that is based on the request context.
- Consists of the required authentication level, and type.

The authentication policy is determined statically based on the configuration or is provided with the query string.

You can modify the static configuration through the Advanced Configuration panel of the administration console. See Chapter 12, “Managing advanced configuration,” on page 107.

To specify the policy with the query string, see “Managing mapping rules” on page 49.

User password attempts

When users attempt to log in using HOTP or TOTP and submit an incorrect one-time password, they receive one *strike* against their account. This strike remains on their account for a configurable duration. By default, the duration is 10 minutes. After that duration, the strike is removed from their account.

When users submit multiple incorrect one-time passwords, they can reach a maximum and are then prevented from making another attempt until one of their strikes expires. By default, the maximum is 5.

If the users log in successfully, any strikes on their account are cleared.

Strikes are shared between TOTP and HOTP. For example, if the users made two incorrect attempts using TOTP, those strikes count against them on HOTP as well.

Because user retries affect only TOTP and HOTP logins, users who exceeded password attempt using those logins can still use other OTP provider logins or basic username/password authentication.

You can modify the password retry settings through the Advanced Configuration panel of the administration console. See Chapter 12, “Managing advanced configuration,” on page 107.

One-time password delivery methods and providers

Security Access Manager provides functionality to generate one-time passwords to users who want to access your protected resources. To deliver the generated passwords to users, you can use short message service (SMS) or email. HTML template pages are provided for you to customize the use, generation, and delivery of the passwords.

You can configure any of the following password providers and delivery methods:

MAC The MAC provider generates one-time passwords by randomly drawing one character at a time from the configured character set until the configured number of characters are drawn. The MAC provider also stores the generated one-time passwords in the configured one-time password store plug-in. Each one-time password is salted and hashed before it is stored in the configured one-time password store plug-in.

TOTP The TOTP provider generates one-time passwords by using a specified algorithm with a time-based one-time password application. Passwords are not communicated or stored, but are verified as a match between server and client as they are regenerated at regular intervals.

HOTP The HOTP provider generates one-time passwords by using a specified algorithm with a counter-based one-time password application. Passwords are not communicated or stored, but are verified as incremental matches between server and client.

RSA The RSA provider works with an RSA SecurID Authentication Manager and passcode generator. You must own the RSA Authentication Manager product to use RSA as a provider. The RSA Authentication Manager and passcode generator generates a passcode every 30 - 60 seconds. The user name and passcode are supplied by the user and passed to the RSA Authentication Manager. The RSA Authentication Manager makes a decision and returns it to Security Access Manager, which relays the decision back to the user.

Email Delivery

Delivers the one-time password by using email. The **Email Delivery** sends the email address of the user and the one-time password in a message, whose MIME type is `text/plain`, to the configured SMTP Server. The SMTP Server then sends the one-time password to the user by email. The product does not provide an SMTP Server. You must configure your own SMTP Server.

SMS Delivery

Delivers the one-time password by using the Short Message Service or SMS. The **SMS Delivery** first sends the phone number of the user and the one-time password in an **HTTP POST** request, whose content type is `application/x-www-form-urlencoded`, to the configured SMS Gateway. The SMS Gateway then sends the one-time password to the user through SMS. The product does not provide an SMS Gateway. You must configure your own SMS Gateway.

Configuring an HOTP one-time password provider

The HOTP one-password provider relies on a public algorithm to generate the one-time password.

About this task

The HOTP client solution and the Security Access Manager use the same algorithm to generate the one-time password value. No interaction is required between the client software and the Security Access Manager solution. The algorithm uses a shared secret key and a counter to generate the one-time password value. Every time a new one-time password is generated, the counter value increments on both server and client solutions. No delivery of the one-time password is required.

This task describes the steps and properties for configuring a HOTP provider. For information about configuring other providers, see:

- “Configuring a MAC one-time password provider” on page 41
- “Configuring a TOTP one-time password provider” on page 40
- “Configuring an RSA one-time password provider” on page 43

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Authentication**.
4. Click **Mechanisms**.
5. Click **HOTP One-time Password**.
6. Complete the properties for the provider.

HOTP

Max Counter Lookahead

The number of times to increment the counter to see whether the one-time password is valid before stopping. Any non-negative number is valid.

The default is 25.

Password Length

The length of the generated one-time passwords, which can be 6 - 9 characters or numbers.

The default is 6.

Generation Algorithm

The algorithm that is used to generate the one-time password. Valid options include the following algorithms:

- HmacSHA1
- HmacSHA256
- HmacSHA512

The default is HmacSHA1.

Secret key URL

The URL that is used to deliver the secret key. The QR code is also generated using this URL. The URL format might include information specific to your environment, such as your company name.

The default URL is:

```
otpauth://hotp/Example:@USER_NAME@?secret=@SECRET_KEY@&issuer=Example&counter=0
```

The URL uses two macros. Place these macros in the URL wherever their corresponding values belong.

@SECRET_KEY@

The secret key.

@USER_NAME@

The user name of the authorized user who logs in.

Attention: You must change this URL to the appropriate URL for your environment. For example:

```
otpauth://hotp/Example:sec_master?secret=4W706TZGEZKN5PUY&issuer=Example&counter=0
```

7. Click **Save**.

8. When you configure one-time password providers, a message indicates that changes have not been deployed. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Configuring a TOTP one-time password provider

The TOTP one-password provider relies on a public algorithm to generate the one-time password.

About this task

The TOTP client solution and the Security Access Manager use the same algorithm to generate the one-time password value. No interaction is required between the client software and the Security Access Manager solution. The algorithm uses a shared secret key and the time to generate the one-time password value. No delivery of the one-time password is required.

This task describes the steps and properties for configuring a TOTP provider. For information about configuring other providers, see:

- “Configuring a MAC one-time password provider” on page 41
- “Configuring an HOTP one-time password provider” on page 38
- “Configuring an RSA one-time password provider” on page 43

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Authentication**.
4. Click **Mechanisms**.
5. Click **TOTP One-time Password**.
6. Complete the properties for the provider.

TOTP

Generation Interval (seconds)

The number of seconds an interval lasts. This number determines how long a one-time password is active before the next one-time password generates.

The default is 30.

Password Length

The length of the generated one-time passwords, which can be 6 - 9 characters or numbers.

The default is 6.

Skew Intervals

The skew intervals of the algorithm. The skew intervals consider any possible synchronization delay between the server and the client that generates the one-time password. For example, a skew interval of 2 means a one-time password in up to two intervals in the past, or two in the future are valid. For example, if it is interval 563, and intervals are 30 seconds, then one-time passwords for intervals 561-565 are computed and checked against within a range of 2.5 minutes.

The default is 1.

One Time Use

Whether to cache one-time passwords if they are used to successfully log in. If set to true, then the reuse of a one-time password is prevented while it is in cache.

The default is true.

Generation Algorithm

The algorithm that is used to generate the one-time password. Valid options include the following algorithms:

- HmacSHA1
- HmacSHA256
- HmacSHA512

The default is HmacSHA1.

Secret key URL

The URL that is used to deliver the secret key. The QR code is also generated using this URL. The URL format might include information specific to your environment, such as your company name.

The default URL is:

```
otpauth://totp/Example:@USER_NAME@?secret=@SECRET_KEY@&issuer=Example
```

The URL uses two macros. Place these macros in the URL wherever their corresponding values belong.

@SECRET_KEY@

The secret key.

@USER_NAME@

The user name of the authorized user who logs in.

Attention: You must change this URL to the appropriate URL for your environment. For example:

```
otpauth://totp/Example:sec_master?secret=4W706TZGEZKN5PUY&issuer=Example
```

7. Click **Save**.
8. When you configure one-time password providers, a message indicates that changes have not been deployed. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Configuring a MAC one-time password provider

A one-time password is valid for one session or login. The MAC password is generated by Security Access Manager and can be delivered to the user through Short Message Service (SMS) or e-mail.

About this task

This task describes the steps and properties for configuring a MAC provider. For information about configuring other providers, see:

- “Configuring an HOTP one-time password provider” on page 38
- “Configuring a TOTP one-time password provider” on page 40
- “Configuring an RSA one-time password provider” on page 43

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Authentication**.
4. Click **Mechanisms**.
5. Click **MAC One-time Password**.
6. Complete the properties for the provider.

MAC

Password Character Set

The character set from which the characters in the one-time password are generated.

The default is 0123456789.

Password Length

The length of the characters in the one-time password.

The default is 8.

Store Entry Hash Algorithm

The hash algorithm that is used for hashing the one-time password before it is stored in the one-time password store plug-in. The supported algorithms are:

- SHA1
- SHA-256
- SHA-512

The default is SHA-256.

Store Entry Lifetime (seconds)

The length of time that the one-time password is stored. The lifetime is in seconds.

The default is 300.

7. Click **Save**.
8. When you configure one-time password providers, a message indicates that changes have not been deployed. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

What to do next

Next, consider configuring the delivery methods for the passwords. Both SMS and Email delivery are enabled but you will want to configure the delivery properties, such as SMTP server or connection URL, for your environment. See “Configuring one-time password delivery methods” on page 46.

Configuring an RSA one-time password provider

A one-time password is valid for one session or login. To use RSA as a provider, you must own RSA Authentication Manager. The server and the client generate the passwords with the same algorithm.

About this task

This task describes the steps and properties for configuring an RSA provider. For information about configuring other providers, see:

- “Configuring an HOTP one-time password provider” on page 38
- “Configuring a MAC one-time password provider” on page 41
- “Configuring a TOTP one-time password provider” on page 40

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Authentication**.
4. Click **Mechanisms**.
5. Click **RSA One-time Password**.
6. Complete the properties.

Agent Network Interface

The name of the network interface that the RSA Agent is using to connect to the RSA server.

Required: Yes

Data type: String

Valid values:

Management network interface values

- M.1
- M.2

Application network interface values

- P.1
- P.2
- P.3
- P.4

Note: If you are using the RSA Provider in a cluster environment and use an application interface with multiple IP addresses defined for that interface, use the RSA console to add all of those IP addresses to the whitelist. See the RSA documentation for information about adding IP addresses to the whitelist.

Example: M.1

Server Exchange Initial Timeout

The initial timeout coefficient in milliseconds used to calculate the timeout of the request.

Required: No

Data type: Integer

Example: 1000

Server Exchange Timeout Offset

The offset timeout coefficient in milliseconds used to calculate the timeout of the request.

Required: No

Data type: Integer

Example: 200

Server Exchange Timeout Increment

The increment coefficient in milliseconds used to calculate the timeout of the request.

Required: No

Data type: Integer

Example: 100

Event Log Level

The minimum event level to be logged. Events below the level that is specified in this property are not logged.

The events in order from lowest level to highest are:

- a. OFF
- b. DEBUG
- c. INFO
- d. WARN
- e. ERROR
- f. FATAL

Required:

Data type: String

Example: INFO. If this property is set to INFO, the DEBUG errors are not logged.

Enable Debug Tracing

The property that enables debug tracing.

Required: No

Data type: Boolean

Example: FALSE. If set to FALSE, debug tracing is not enabled.

Trace Function Entries

The property that enables tracing of function entries.

Required: No

Data type: Boolean

Example: FALSE. If set to FALSE, function entries are not traced.

Trace Function Exits

The property that enables tracing of exits.

Required: No

Data type: Boolean

Example: FALSE. If set to FALSE, exits are not traced.

Trace Flow Statements

The property that enables tracing of flow statements.

Required: No

Data type: Boolean

Example: FALSE. If set to FALSE, flow statements are not traced.

Trace Regular Statements

The property that enables tracing of regular statements.

Required: No

Data type: Boolean

Example: FALSE. If set to FALSE, regular statements are not traced.

Trace Location

The property that enables the class name and line number to be displayed in the trace.

Required: No

Data type: Boolean

Example: FALSE. If set to FALSE, class name and line number are not displayed.

7. Click **Save**.
8. On your RSA server, generate the following files:

sdconfs.rec

The configuration file for connecting to the RSA Authentication server.

sdopts.rec

The configuration properties file that contain optional configurations for load balancing.

securid

The secret key file for connecting to the RSA Authentication server.

See your RSA Authentication server documentation for details on creating these files and use the following guidelines:

- On the appliance, you must specify an Agent Network Interface. See Agent Network Interface in step 6 on page 43. If you connect the RSA server to the appliance using an application network interface with multiple IP addresses, list all the IP addresses in the **Alternate IPs** box on the RSA server.
- For **Agent type**, choose **Standard**.
- **Agent Auto-Registration** must be enabled when the first RSA one-time password authentication is performed. You can disable it after the first successful authentication is completed.

Note: The RSA one-time password provider does not support replication of the RSA session information through the cluster environment. The session information is local to each cluster node and the environment should be configured to enforce session affinity between the client and the cluster node.

9. Move or copy the generated files from the RSA server to the appliance.
10. Log in to the local management interface.
11. Click **Secure Mobile Settings**.
12. Under **Policy**, click **Authentication**.

13. Click **Mechanisms**.
14. Click **RSA One-time Password**.
15. Select a file in the table that corresponds to the file you generated on the RSA server.
16. Click **Upload** to upload the file or **Clear** to remove the contents of the selected file. The status area indicates one of three statuses:

Not uploaded

Upload is not completed.

Last upload date

Upload was completed on date indicated.

Auto-generated

The SecurID was automatically generated instead of uploaded.

Repeat this step until all of your files have been uploaded to the appliance.

17. Click **Save**.
18. When you configure one-time password delivery methods or providers, a message indicates that changes have not been deployed. If you have finished making changes, deploy them.
For more information, see Chapter 13, "Deploying pending changes," on page 119.

Configuring one-time password delivery methods

Passwords can be delivered to the user through Short Message Service (SMS) or email.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Authentication**.
4. Click **Mechanisms**.
5. Click the delivery type.
 - **SMS One-time Password**
 - **Email One-time Password**
6. Complete the properties for the delivery method.

SMS

Basic Authentication User Name

The user name that is used in HTTP Basic authentication.

SMS Delivery does not perform the HTTP basic authentication if this configuration is not specified.

Required: False

Multi-value: No

Example: username

Basic Authentication Password

The password that is used in HTTP basic authentication.

SMS Delivery does not perform HTTP Basic authentication if this configuration is not specified.

Required: False

Multi-value: No

Example: password

Connection URL

The URL of the SMS Gateway where the phone number of the user and the one-time password is sent.

Required: True

Multi-value: No

Example: `https://msgateway.tfim.example.com/`

HTTP Request Parameters

The list of name and value pairs that is included in the body of the **HTTP POST** request to the SMS Gateway. In each pair, the name and the value must be separated by equal sign.

Two macros, **\$DEST_NO\$** and **\$MSG\$**, are replaced by the phone number of the user and the content of the SMS. These two macros can be used only as value in the name and value pair.

Required: True

Multi-value: Yes

Example:

- `From=+0123456789`
- `To= $DEST_NO$`
- `Body= MSG`

Success HTTP Response Body Regex Pattern

This parameter defines the Java™ regular-expression pattern that matches the HTTP response body that is returned by the SMS Gateway. When the match is successful, the SMS delivery is successful.

The default value is empty.

The default behavior is that the HTTP response body is not going to be matched against any Java regular-expression and the success or failure decision is going to be based on the `SuccessHTTP`

`ReturnCode` value only.

Note: If the HTTP response from the SMS Gateway does not contain a body, this matching is not performed.

Required: False

Multi-value: No

Example:

- When the body of all responses by the SMS Gateway contains either `Success` or `Failure` followed by no newline character, the sample `SuccessHTTP`
Response
BodyRegex
Pattern value is
`Success`

- When the body of all responses by the SMS Gateway contains the following text:

```
MGDID=TTTT
TTTTTTTT
RESPONSE
CODE=NNN
SMS=TTTTTTT
TTTTTTTT
TTTTTTT
DATE=NNNNNNNN
```

where each line ends with the \n character without any preceding \r character, and the RESPONSECODE is defined such that a three-digit number from 0 to 199 indicates success, the sample SuccessHTTP

ResponseBody
RegexPattern value is

```
(?s).*
RESPONSE
CODE=(\d{1,2}
|[0-1]{1}
\d{2})\n.*
```

Success HTTP Return Code

The response code from the SMS Gateway that is an acknowledgment from the SMS Gateway that the request is successfully processed.

The default SuccessHTTP ReturnCode, which is 200, is used when this configuration is not specified.

Note: The SuccessHTTP ReturnCode match must be successful before this matching is done.

Required: False

Multi-value: No

Example: 200

HTTPS Trust Store

The keystore that validates the SMS Gateway SSL certificate.

This configuration must be specified only when SMS Delivery communicates with the SMS Gateway by using HTTPS.

Required: False

Multi-value: No

Example: DefaultTrustedStore

Client Authentication Key

The certificate that is used as client certificate in SSL Client authentication. The certificate is a keystore and alias pair. The keystore and alias must be separated by underscore.

SMS Delivery does not perform SSL Client authentication if this configuration is not specified.

Required: False

Multi-value: No

Example: DefaultKeystore_testkey

Email

Sender Email

The email address that is used as the sender of the email that is sent to the user.

Required: True

Multi-value: No

Example: otp_emailer@example.com

SMTP Host Name

The host name of the SMTP Server.

Required: True

Multi-value: No

Example: smtpserver.tfim.example.com

SMTP User Name

The user name that is used in SMTP authentication.

Required: False

Multi-value: No

Example: username

SMTP Password

The password that is used in SMTP authentication.

Required: False

Multi-value: No

Example: password

7. Click **Save**.
8. When you configure one-time password delivery methods, a message indicates that changes have not been deployed. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.


Managing mapping rules

The one-time password mapping rules are JavaScript code that run during the one-time password flow. Use the rules to customize the provider and delivery methods to use for the one-time password flow and to customize how the one-time password is generated, delivered, and verified.


Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Authentication**.
4. Click **Advanced**.
5. Take one of the following actions:

View a mapping rule:


- a. Select a mapping rule.
- b. Click . The View Mapping Rule panel opens. The content of the mapping rule is displayed.
- c. Click **OK** to close the panel.

Export a mapping rule:

- a. Select a mapping rule.
- b. Click .
- c. Choose a location and save the file.

Replace a mapping rule:

Note: Use an existing mapping rule as the basis for the updated mapping rule.

- a. Select a mapping rule that you want to replace.
- b. Click .
- c. Click the field or the **Browse** button and select a file.

Attention: The name of the mapping rule cannot be replaced. The name of the uploaded file is ignored.

- d. Click **OK** to upload the mapping rule.

6. When you replace a mapping rule, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

OTPGetMethods mapping rule

OTPGetMethods specifies the methods for delivering the one-time password to the user.

This sample mapping rule sets password delivery conditions for the following delivery methods:

- By email
- By SMS
- No delivery

Each delivery method includes the following attributes and their corresponding value:

id Specifies a unique delivery method ID. This value replaces the @OTP_METHOD_ID@ macro in the OTP Method Selection page. Use a unique value across different methods. For example, sms.

deliveryType

Specifies the delivery plug-in that delivers the one-time password. The value must match one of the types in the DeliveryTypesToOTPDeliveryModuleIds parameter of the OTP response file. For example, sms_delivery.

deliveryAttribute

Specifies an attribute that is associated with the delivery type. The value depends on the one-time password provider plug-in for the delivery type. For example:

- For SMS delivery, the value is the mobile number of the user. For example, `mobileNumber`.
- For email delivery, the value is the email address of the user. For example, `emailAddress`.
- For no delivery, the value is an empty string.

label Specifies the unique delivery method to the user. For time-based and counter-based one-time password, use this attribute to specify the secret key of the user. If `label` is not specified, the time-based and counter-based one-time password code retrieves the key by invoking the user information provider plug-in. This parameter replaces the `@OTP_METHOD_LABEL@` macro in the OTP Method Selection page.

otpType

Specifies the one-time password provider plug-in that generates and verifies the password. The value must match one of the types in the `OTPTypesToOTPProviderModuleIds` parameter of the OTP response file. For example, `mac_otp`.

userInfoType

Specifies which user information provider plug-in to use to retrieve user information that is required to calculate the one-time password. This parameter is only required if user information is used for calculation of the one-time password.

To customize one-time password delivery, you can do one of the following actions:

- Create your own mapping rules that are based on the sample `OTPGetMethods` mapping rule.
- Modify the sample `OTPGetMethods` mapping rule.

You can also customize the mapping rule to use access control context data. For details see, “Customizing one-time password mapping rules to use access control context data” on page 54.

OTPGenerate mapping rule

`OTPGenerate` mapping rule specifies the generation of the one-time password for the user.

You can use the `OTPGenerate` mapping rule in the following configuration:

Modify the one-time password type of the selected method to generate the one-time password

Indicates the one-time password type to determine the one-time password Provider plug-in that generates the one-time password for the user.

Note: See the comments in the mapping rule for more details.

You can also customize the mapping rule to use access control context data. For details see, “Customizing one-time password mapping rules to use access control context data” on page 54.

OTPDeliver mapping rule

The `OTPDeliver` mapping rule specifies the delivery method of the one-time password to the user.

Use the following `OTPDeliver` mapping rules:

Generate the one-time password hint

The one-time password hint is a sequence of characters that is associated with the one-time password. The one-time password hint is displayed in the One-Time Password Login page. It is also sent to the user together with the one-time password.

You can customize the way the one-time password hint is generated by modifying the following section in the default OTPDeliver mapping rule:

```
var otpHint = Math.floor(1000 + (Math.random() * 9000));
```

Note: See the comments in the mapping rule for more details.

Generate the formatted one-time password

The formatted one-time password is the formatted version of the one-time password. The formatted one-time password, instead of the actual one-time password, is sent to the user. For example, for one-time password hint abcd, and one-time password 12345678, you can set the formatted one-time password as abcd-12345678. For one-time password hint efgh, and one-time password87654321, you can set the one-time password as efgh#8765#4321.

You can customize the way that the one-time password is generated by modifying the following section in the sample OTPDeliver mapping rule:

```
var otpFormatted = otpHint + "-" + otp;
```

Note: See the comments in the mapping rule for more details.

Modify the delivery type of the selected method for delivering the one-time password

The delivery type specifies the one-time password Delivery plug-in that delivers the one-time password to the user.

Modify the delivery attribute of the selected method to deliver

The delivery attribute is an attribute that is associated with delivery type. The meaning of the delivery attribute depends on the one-time password provider plug-in for the delivery type. For example, for SMS delivery type, the delivery attribute is the mobile number of the user. For email delivery type, the delivery attribute is the email address of the user.

Note: See the comments in the mapping rule for more details.

You can also customize the mapping rule to use access control context data. For details see, "Customizing one-time password mapping rules to use access control context data" on page 54.

OTPVerify mapping rule

OTPVerify specifies the verification of the one-time password that is submitted by the user.

You can customize the sample OTPVerify mapping rule to modify the following verification rules:

Modify the one-time password type of the user

Indicates the one-time password type to determine the one-time Provider plug-in that verifies the one-time submitted by the user.

Set the authentication level of the user

After one-time password authentication completes, a credential is issued

that contains the authentication level of the user. You can customize the authentication level by modifying the following section in the mapping rule:

```
var authenticationLevel = contextAttributesAttributeContainer.getAttributeValueByNameAndType
("otp.otp-callback.authentication-level", "otp.otp-callback.type");
var attributeAuthenticationLevel = new Attribute("AUTHENTICATION_LEVEL",
"urn:ibm:names:ITFIM:5.1:accessmanager", authenticationLevel);
attributeContainer.setAttribute(attributeAuthenticationLevel);
```

Enforce the number of times the user can submit the one-time password in the one-time password login page

If a user exceeds the permitted number of times to submit a one-time password, an error message displays. You can customize the number of times that the user can submit the one-time password in the one-time password login page by modifying the following section in the mapping rule:

```
var retryLimit = 5;
```

By default, this option is set to false.

Note: This setting applies only to MAC OTP.

Identify the secret key of a user

When a user registers with a time-based one-time password application, they are assigned a secret key. Store the secret key in this mapping rule for verification of the user by modifying the following code:

```
var secretStr = new java.lang.String(SECRET_KEY_GOES_HERE);
```

By default, this option is set to false.

Override the one-time password target URL

By default, a user is redirected to a target URL upon completion of an one-time password flow. That target URL was either the initial cached request at the WebSEAL or reverse proxy instance or was specified as part of the one-time password invocation using the **Target** query string parameter.

You can use the `OTPVerify` mapping rule to override this target URL by adding an attribute called `itfim_override_targeturl_attr`. This attribute ensures that at the completion of a successful one-time password flow, the user is redirected to the override target instead of the initial target.

Example code:

```
var targetUrl = new java.lang.String("http://www.example.com/url");
var targetUrlAttr = new Attribute("itfim_override_targeturl_attr",
"urn:ibm:names:ITFIM:5.1:accessmanager", targetUrl);
attributeContainer.setAttribute(targetUrlAttr);
```

To customize one-time password verification, you can do one of the following actions:

- Create your own verification rules that are based on the sample `OTPVerify` mapping rule.
- Modify the sample `OTPVerify` mapping rule.

You can also customize the mapping rule to use access control context data. For details see, “Customizing one-time password mapping rules to use access control context data” on page 54.

Customizing one-time password mapping rules to use access control context data


Some authentication scenarios require that context data used in making an access control decision be available during authentication. You can configure Security Access Manager to capture the content data and make it available to the one-time password mapping rules.

About this task

You can configure Security Access Manager to perform access control policy evaluation when a resource is accessed. The access control policy evaluation can result on a permit with authentication. The required authentication is determined by the access control policy. Some scenarios require that the context data used to perform the access control decision be available during the authentication. In order to provide access to the access control context data, you can persist the context information for the predefined authentication obligations that perform one-time password authentication.

Note: The context data available is limited to the attributes referenced by the access control policy and the request attributes provided by the policy enforcement point. If the policy relies on the risk score to perform access control, the context data available also includes the risk-profile attributes.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings > Manage > Advanced Configuration**.
3. Select **attributeCollection.authenticationContextAttributes**.
4. Click  for the property.
5. In the text field, enter a list of comma separated attribute names to be collected during the authorization policy evaluation. For example, if your scenario requires the authentication level and host of the request the configuration property, enter `authenticationLevel, http:host`. The access control context data is provided to the one-time password mapping rules as context attributes values. The following format is used:

```
<stsuser:Attribute name="AttributeName-AttributeURI"
  type="authn.service.context.attribute.type.AttributeDatatype">
<stsuser:Value>AttributeValue</stsuser:Value>
</stsuser:Attribute>
```

Where:

- name is the attribute name and attribute identifier separated by a dash (-).
- type is the attribute data type prefixed by `authn.service.context.attribute.type`.

For example the `authenticationLevel` attribute value is added as:

```
<stsuser:Attribute name="authenticationlevel-urn-ibm:
  security:subject:authenticationlevel"
  type="authn.service.context.attribute.type.Integer">
<stsuser:Value>1</stsuser:Value>
</stsuser:Attribute>
```

6. Click **OK**.
7. When you edit a property, a message indicates that there are undeployed changes. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

8. Configure the mapping rule to use the information collected by this property as the context attribute.
 - a. Click **Secure Mobile Settings**.
 - b. Under **Policy**, click **Authentication**.
 - c. Click **Advanced**.
 - d. Select and export the mapping rule.
 - e. Use a text editor and modify the rule to access the attributes collected during the access control policy evaluation in the following format:


```
var accessControlAttribute =
contextAttributesAttributeContainer.getAttributeValueByNameAndType
("AttributeName-AttributeURI",
"authn.service.context.attribute.type.AttributeDatatype");
```

 Where:
 - name is the attribute name and attribute identifier separated by a dash (-).
 - type is the attribute data type prefixed by `authn.service.context.attribute.type`.
 For example, the `authenticationLevel` attribute can be obtained using the following information:


```
var accessControlAuthenticationLevel =
contextAttributesAttributeContainer.getAttributeValueByNameAndType
("authenticationlevel-urn-ibm:security:subject:authenticationlevel",
"authn.service.context.attribute.type.Integer");
```
 - f. Save the mapping rule and take note of its location.
 - g. In the local management interface, click **Secure Mobile Settings**.
 - h. Under **Policy**, click **Authentication**.
 - i. Click **Advanced**.
 - j. Select the mapping rule you want to replace.
 - k. Click **Replace**. The Replace Mapping Rule panel opens.
 - l. Click the field or the **Browse** button and select the file for your saved mapping rule.

Attention: The name of the mapping rule cannot be replaced. The name of the uploaded file is ignored.
 - m. Click **OK** to upload the mapping rule.

One-time password delivery and user customization

To customize the way your users receive and manage one-time passwords, you can edit the provided one-time password template pages or use the REST services.

Modifying one-time password template files

Use the local management interface to manage files and directories in the template files.

About this task

Table 4. One-time password template files

| File name | Description and link to file contents |
|--------------------------------|--|
| otp/change_pin.html | This template file enables the user to enter a new pin. |
| otp/delivery/email_message.xml | This template page is used by <code>EmailOTPDelivery</code> as the content of the email that it sends to the user. |

Table 4. One-time password template files (continued)

| File name | Description and link to file contents |
|--|--|
| otp/delivery/sms_message.xml | This template page is used by SMSOTPDelivery as the content of the SMS that it sends to the user. |
| otp/delivery_selection.html | This template page displays the list of methods for generating, delivering, and verifying the one-time password. |
| otp/errors/allerror.html | This template page displays general errors that happen during the one-time password flow. |
| otp/errors/error_could_not_validate_otp.html | This template page displays errors during the validation of the one-time password that the user submits. |
| otp/errors/error_generating_otp.html | This template page displays errors during the generation of a one-time password. |
| otp/errors/error_get_delivery_options.html | This template page displays errors during the retrieval of the list of methods for delivering one-time password to the user. |
| otp/errors/error_otp_delivery.html | This template page displays errors during the delivery of a one-time password to the user. |
| otp/errors/error_sts_invoke_failed.html | This template page displays errors during the invocation of the Security Token Service. |
| otp/login.html | This template page displays the form where the user can enter the one-time password. |
| otp/next_otp.html | This template page enables the user to enter the next one time password. |

You can run the following tasks on the template files:

- **Edit**- Use this option if you want to view or modify the template file.
- **Import**- Use this option if you to import a file to the template files root.
- **Export**- Use this option if you want to export a file from the template files root.
- **Import Zip**- Use this option if you want to import the template files from a compressed file.
- **Export Zip**- Use this option if you want to export the template files as a compressed file.

Procedure

1. From the top menu, select **Secure Mobile Settings > Manage > Template Files**.
2. Work with all the management files and directories.

View or update the contents of a file in the template files root

- a. Select the file of interest.
- b. Select **Edit**. You can then view the contents of the file.
- c. Edit the contents of the file.
- d. Click **Save**.

Export a file from the template files root

- a. Select the file of interest.

- b. Select **Manage > Export**.
- c. Confirm the save operation when your browser displays a confirmation window.

Import a file to the template files root

- a. Select the file of interest.
- b. Select **Manage > Import**.
- c. Click **Browse**.
- d. Browse to the file you want to import.
- e. Click **Open**.
- f. Click **Import**.

Export the template file as a compressed file

- a. Select **Manage > Export Zip**.
- b. Confirm the save operation when your browser displays a confirmation window.

Import the template files as a compressed file

Make sure that the .zip file contains files that exist in the document root.

- a. Select **Manage > Import Zip**.
- b. Click **Browse**.
- c. Browse to the file you want to import.
- d. Click **Open**.
- e. Click **Import**.

3. When you edit or import template files, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

One-time password template page for login:

This template page displays the form where the user can enter the one-time password. The page is also called One-Time Password Login page.

Macros

The template has the following replacement macros:

@ERROR_MESSAGE@

This macro is replaced with a message that indicates that the submitted one-time password contains errors. Examples of these errors are that the submitted one-time password is not valid and the one-time password is submitted after it expires.

@MAPPING_RULE_DATA@

If the submitted one-time password contains an error, this macro is replaced with the value of the STS Universal User context attribute whose name is @MAPPING_RULE_DATA@ and type is otp.sts.macro.type. This context attribute can be set in the “OTPVerify mapping rule” on page 52.

@OTP_HINT@

This macro is replaced with one-time password hint. The one-time password hint is a sequence of characters that is associated with the one-time password.

@REGENERATE_ACTION@

This macro is replaced with the URL where the **Generate** button posts the form to regenerate and deliver the new one-time password value.

@RESELECT_ACTION@

This macro is replaced with the URL where the Reselect button posts the form to reselect the method for generating, delivering, and verifying the one-time password value.

@OTP_METHOD_TYPE@

This macro is replaced by the type of the currently selected method for generating, delivering, and verifying the one-time password. This type is generated by OTPGetMethods mapping rule and was selected by the user.

Reselect button

This one-time password login page enables a user to select a one-time password provider. By default, the page is designed so that if the user wants to change the selection, the user has to use the **Back** button on the browser. However, a Reselect button can be implemented on the page.

To enable the button, uncomment the portion of the login.html file that enables the button. A comment in the file explains the purpose of the button and how to enable it.

Note: If you add the button to the page but the policy is enabled for a specific provider, such as Permit with authentication HOTP One-time Password, the button appears on the page but is inoperable and takes no action.

One-time password template page for delivery selection:

This template page displays the list of methods for generating, delivering, and verifying the one-time password. It is also called One-Time Password Method Selection page.

@OTP_METHOD_ID@

This macro is replaced by the ID of the method for generating, delivering, and verifying the one-time password. This ID is generated by OTPGetMethods mapping rule.

@OTP_METHOD_LABEL@

This macro is replaced by the label of the method for generating, delivering, and verifying the one-time password. This label is generated by OTPGetMethods mapping rule.

@OTP_METHOD_CHECKED@

For the first method, this macro is replaced with an HTML radio button attribute that causes that radio button to be selected. For the remaining methods for generating, delivering, and verifying, this macro is replaced with an empty string.

One-time password template page for general errors:

This template page displays general errors that happen during the one-time password flow. General errors are errors that are not displayed in other template pages.

The template has the following replacement macros:

@REQ_ADDR@

This macro is replaced with the URL into which the request from the user is sent.

@TIMESTAMP@

This macro is replaced with the timestamp when the error occurred.

@DETAIL@

This macro is replaced with the error message.

@EXCEPTION_STACK@

This macro is replaced with the stack trace of the error.

Figure 2. Template for `allerror.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>One-Time Password Error</title>
</head>
<body style="background-color: #ffffff">
  <div>
    <h2 style="color: #ff8800">An error has occurred.</h2>
    <div id="infoDiv" style="background-color: #ffffff; color: #000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em> <br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
      border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color: #999999;
      border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

One-time password template page for generating one-time password error:

This template page displays errors during the generation of a one-time password.

The template has the following replacement macros:

@REQ_ADDR@

This macro is replaced with the URL into which the request from the user is sent.

@TIMESTAMP@

This macro is replaced with the timestamp when the error occurred.

@DETAIL@

This macro is replaced with the error message.

@EXCEPTION_STACK@

This macro is replaced with the stack trace of the error.

Figure 3. Template for `error_generating_otp.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>One-Time Password Error</title>
</head>

<body style="background-color: #ffffff">
  <div>
    <h2 style="color:#ff8800">An error occurred while
      generating the one-time password.</h2>
    <div id="infoDiv" style="background-color: #ffffff; color: #000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em><br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
      border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color: #999999;
      border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

One-time password template page for get delivery error:

This template page displays errors during the retrieval of the list of methods for delivering one-time password to the user.

The template has the following replacement macros:

@REQ_ADDR@

This macro is replaced with the URL into which the request from the user is sent.

@TIMESTAMP@

This macro is replaced with the timestamp when the error occurred.

@DETAIL@

This macro is replaced with the error message.

@EXCEPTION_STACK@

This macro is replaced with the stack trace of the error.

Figure 4. Template for `error_get_delivery_options.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>One-Time Password Error</title>
</head>

<body style="background-color: #ffffff">
  <div>
    <h2 style="color: #ff8800">An error occurred while
    obtaining the one-time password delivery options.</h2>
    <div id="infoDiv" style="background-color: #ffffff; color: #000000">
      <em>@REQ_ADDR@</em> <br />
      <em>@TIMESTAMP@</em> <br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
    border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color: #999999;
    border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

One-time password template page for delivery error:

This template page displays errors during the delivery of a one-time password to the user.

The template has the following replacement macros:

@REQ_ADDR@

This macro is replaced with the URL into which the request from the user is sent.

@TIMESTAMP@

This macro is replaced with the timestamp when the error occurred.

@DETAIL@

This macro is replaced with the error message.

@EXCEPTION_STACK@

This macro is replaced with the stack trace of the error.

Figure 5. Template for error_otp_delivery.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color:#ff8800">An error occurred while
    delivering the one-time password value.</h2>
    <div id="infoDiv" style="background-color: #ffffff; color: #000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em><br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
    border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color: #999999;
    border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

One-time password template page for security trust service operation error:

This template page displays errors during the invocation of the Security Token Service.

The template has the following replacement macros:

@REQ_ADDR@

This macro is replaced with the URL into which the request from the user is sent.

@TIMESTAMP@

This macro is replaced with the timestamp when the error occurred.

@DETAIL@

This macro is replaced with the error message.

@EXCEPTION_STACK@

This macro is replaced with the stack trace of the error.

Figure 6. Template for error_sts_invoke_failed.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
```

```

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>One-Time Password Error</title>
</head>

<body style="background-color: #ffffff">
  <div>
    <h2 style="color: #ff8800">An error occurred while
    invoking the trust service to perform a one-time password operation.</h2>
    <div id="infoDiv" style="background-color: #ffffff; color: #000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em><br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
    border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color: #999999;
    border-style: solid; border-width: 1px; border-color: #000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>

```

One-time password template page for one-time password validation error:

This template page displays errors during the validation of the one-time password that the user submits.

The template has the following replacement macros:

@REQ_ADDR@

This macro is replaced with the URL into which the request from the user is sent.

@TIMESTAMP@

This macro is replaced with the timestamp when the error occurred.

@DETAIL@

This macro is replaced with the error message.

@EXCEPTION_STACK@

This macro is replaced with the stack trace of the error.

Figure 7. Template for `error_could_not_validate_otp.html`

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color: #ff8800">The one-time password
    value could not be validated.</h2>

```

```

<div id="infoDiv" style="background-color: #ffffff; color: #000000">
  <em>@REQ_ADDR@</em> <br />
  <em>@TIMESTAMP@</em> <br />
</div>
<br />
<div id="detailDiv" style="background-color: #999999;
border-style: solid; border-width: 1px; border-color: #000000">
  <h4>Error details</h4>
  @DETAIL@
</div>
<br />
<div id="stackDiv" style="background-color: #999999;
border-style: solid; border-width: 1px; border-color: #000000">
  <h4>Stack trace</h4>
  @EXCEPTION_STACK@
</div>
</div>
</body>
</html>

```

One-time password template page for Short Message Service (SMS):

This template page is used by SMSOTPDelivery as the content of the SMS that it sends to the user.

The template has the following replacement macro:

@OTP_STRING@

This macro is replaced with the one-time password generated by the one-time password provider plug-in.

Figure 8. Template page for sms_message.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
<Message>
  <Value>
    This is your one-time password @OTP_STRING@.

    Thank you,
    OTP Test
  </Value>
</Message>
</root>

```

One-time password template page for email:

This template page is used by EmailOTPDelivery as the content of the email that it sends to the user.

The template has the following replacement macro:

@OTP_STRING@

This macro is replaced with the one-time password generated by the one-time password provider.

Figure 9. Template for email_message.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
<Subject>
  <Value>
    One-time password
  </Value>
</Subject>

<Message>
  <Value>
    This is your one-time password @OTP_STRING@.

    Thank you,
    OTP Test
  </Value>
</Message>
</root>

```

REST services for OTP secret keys

You can use the REST services capability to help manage your mobile data, such as your HOTP and TOTP secret keys.

To help prevent unauthorized users from confiscating and resetting the secrets keys that belong to authorized users, the administrator must complete the following steps:

1. Write a policy that requires a form of two-factor authentication other than the following authentication types:
 - HOTP
 - TOTP
2. Attach the policy to the OTP management URLs.

Note: The user must authenticate to use the REST services capability.

REST services usage scenarios

Depending on your usage scenario, type the following URLs into the web page that calls the REST services:

Table 5. REST services instructions

| Method | URL | Response | Response type |
|--------|---|--|------------------|
| GET | <p><code>https://hostname/mga/sps/mga/user/mgmt/otp/{otpType}</code></p> <p>Note: Valid values for <i>otpType</i> include the following values:</p> <ol style="list-style-type: none"> 1. totp 2. hotp | <pre> {"username": username, "secretKey": secretKey, "secretKeyUrl": secretKeyUrl} </pre> <p>If the request completes successfully, the HTTP response code is 200.</p> <p>If the request does not complete successfully, the HTTP response is 500.</p> | application/json |

Table 5. REST services instructions (continued)

| Method | URL | Response | Response type |
|--------|---|---|------------------|
| GET | <p>https://hostname/mga/sps/mga/user/mgmt/otp/qr/{otpType}</p> <p>Note: Valid values for <i>otpType</i> include the following values:</p> <ol style="list-style-type: none"> 1. totp 2. hotp | <p>Quick response (QR) code</p> <p>If the request completes successfully, the HTTP response code is 200.</p> <p>If the request does not complete successfully, the HTTP response is 500.</p> | image/gif |
| DELETE | <p>https://hostname/mga/sps/mga/user/mgmt/otp/{otpType}</p> <p>Note: Valid values for <i>otpType</i> include the following values:</p> <ol style="list-style-type: none"> 1. totp 2. hotp | <p>{"result": message}</p> <p>If the request completes successfully, the HTTP response code is 200.</p> <p>If the request does not complete successfully, the HTTP response is 500. The following message is in the JSON response: FBTRBA168E The HMAC OTP secret key could not be reset.</p> | application/json |

Chapter 10. OAuth 2.0 support

IBM Security Access Manager for Mobile supports the OAuth 2.0 protocol. The implementation of OAuth 2.0 in Security Access Manager for Mobile strictly follows the OAuth 2.0 standards.

OAuth is an HTTP-based authorization protocol. It gives third-party applications scoped access to a protected resource on behalf of the resource owner. It gives scoped access by creating an approval interaction between the resource owner, client, and the resource server. It gives users the ability to share their private resources between sites without providing user names and passwords. Private resources can be anything, but common examples include photos, videos, contact lists, and so on.

For a complete description of the OAuth 2.0 specifications, see the OAuth website: <http://www.oauth.net>.

OAuth 2.0 concepts

This topic introduces the main concepts of OAuth 2.0.

The following concepts are generally used in OAuth 2.0.

Resource owner

An entity capable of authorizing access to a protected resource. When the resource owner is a person, it is called an *end user*.

OAuth client

A third-party application that wants access to the private resources of the resource owner. The OAuth client can make protected resource requests on behalf of the resource owner after the resource owner grants it authorization. OAuth 2.0 introduces two types of clients: confidential and public. Confidential clients are registered with a client secret, while public clients are not.

OAuth server

Known as the **Authorization server** in OAuth 2.0. The server that gives OAuth clients scoped access to a protected resource on behalf of the resource owner. The server issues an access token to the OAuth client after it successfully does the following actions:

- Authenticates the resource owner.
- Validates a request or an authorization grant.
- Obtains resource owner authorization.

An authorization server can also be the resource server.

Access token

A string that represents authorization granted to the OAuth client by the resource owner. This string represents specific scopes and durations of access. It is granted by the resource owner and enforced by the OAuth server.

Protected resource

A restricted resource that can be accessed from the OAuth server using authenticated requests.

Resource server

The server that hosts the protected resources. It can use access tokens to accept and respond to protected resource requests. The resource server might be the same server as the authorization server.

Authorization grant

A grant that represents the resource owner authorization to access its protected resources. OAuth clients use an authorization grant to obtain an access token. There are four authorization grant types: authorization code, implicit, resource owner password credentials, and client credentials.

Authorization code

A code that the Authorization server generates when the resource owner authorizes a request.

Refresh token

A string that is used to obtain a new access token.

A refresh token is optionally issued by the authorization server to the OAuth client together with an access token. The OAuth client can use the refresh token to request another access token that is based on the same authorization, without involving the resource owner again.

OAuth 2.0 endpoints

Endpoints provide OAuth clients the ability to communicate with the OAuth server or authorization server within a definition.

All endpoints can be accessed through URLs. The syntax of the URLs is specific to the purpose of the access.

If you are responsible for installing and configuring Security Access Manager for Mobile, you might find it helpful to be familiar with these endpoints and URLs.

API protection definitions

The API protection definitions naming follows the standard Security Access Manager for Mobile naming convention. The syntax is:

`https://<hostname:port>/<junction>/sps/oauth/oauth20`

For example:

`https://server.oauth.com/mga/sps/oauth/oauth20`

The following table describes the endpoints that are used in an API protection definition.

Notes:

- There is only a single set of endpoints.
- Not all authorization grant types use all three endpoints in a single OAuth 2.0 flow.

Table 6. OAuth 2.0 endpoint definitions and URLs

| Endpoint name | Description | Example |
|------------------------|--|---|
| Authorization endpoint | An authorization URL where the resource owner grants authorization to the OAuth client to access the protected resource. | <code>https://server.oauth.com/mga/sps/oauth/oauth20/authorize</code> |

Table 6. OAuth 2.0 endpoint definitions and URLs (continued)

| Endpoint name | Description | Example |
|---|--|---|
| Token endpoint | A token request URL where the OAuth client exchanges an authorization grant for an access token and an optional refresh token. | https://server.oauth.com/mga/sps/oauth/oauth20/token |
| Clients manager endpoint | <p>A URL for resource owners to manage their trusted clients.</p> <p>The resource owner can use the clients manager endpoint to access and modify the list of clients that are authorized to access the protected resource. The trusted clients manager shows the client name and permitted scope of an authorized client.</p> <p>Note: The list does not show clients that are disabled or deleted from the definition.</p> <p>The resource owner can optionally remove trusted client information from the list. In doing so, the resource owner is prompted for consent to authorize the next time the OAuth client attempts to access the protected resource.</p> | https://server.oauth.com/mga/sps/oauth/oauth20/clients |
| Session endpoint | <p>A URL where an access_token can be exchanged for a web session. The client uses the endpoint to obtain an authenticated web session for the resource owner that is typically used in hybrid mobile application scenarios.</p> <p>Note: The session endpoint is disabled by default and can be enabled by using advanced configuration. The client must send a POST request with the access_token in the body.</p> <pre>POST /mga/sps/oauth/oauth20/session HTTP/1.1Host: server.oauth.com Content-Type: application /x-www-form-urlencoded access_token=abc123...</pre> | https://server.oauth.com/mga/sps/oauth/oauth20/session |
| Authorization grant management endpoint | A URL where you can view your authorization grants and the tokens and attributes of each authorization grant. | http://server.oauth.com/mga/sps/mga/user/mgmt/html/device/device_selection.html |

OAuth 2.0 workflow

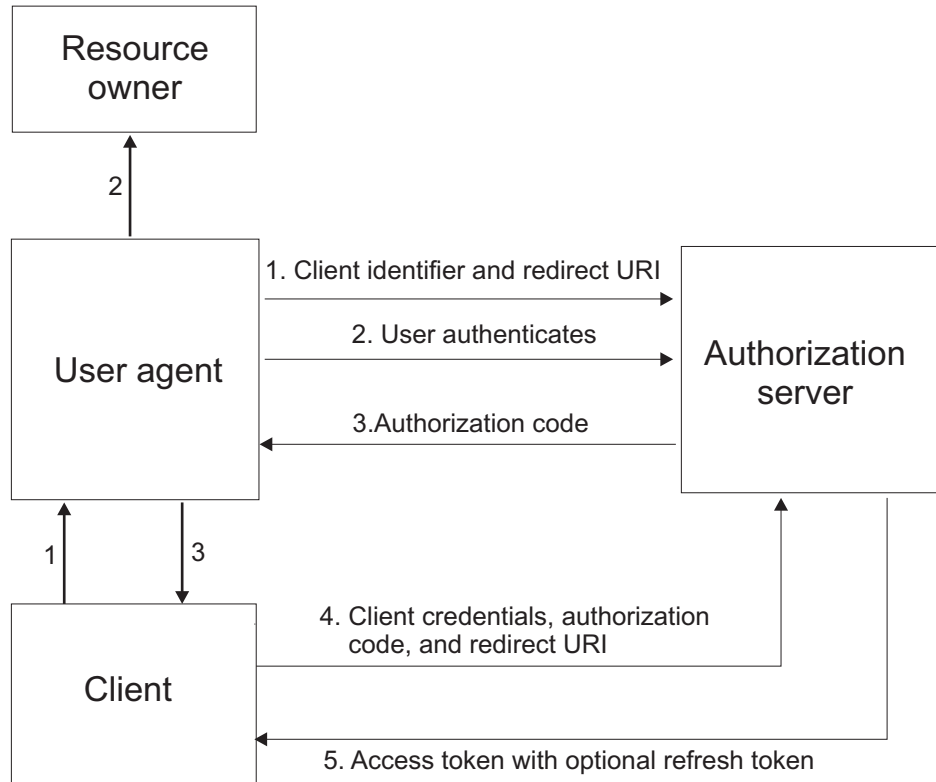
The OAuth 2.0 support in IBM Security Access Manager provides four different ways for an OAuth client to obtain access the protected resource.

OAuth 2.0 workflow

Security Access Manager for Mobile supports the following OAuth 2.0 workflows.

Authorization code flow

The authorization code grant type is suitable for OAuth clients that can keep their client credentials confidential when authenticating with the authorization server. For example, a client implemented on a secure server. As a redirection-based flow, the OAuth client must be able to interact with the user agent of the resource owner. It also must be able to receive incoming requests through redirection from the authorization server.

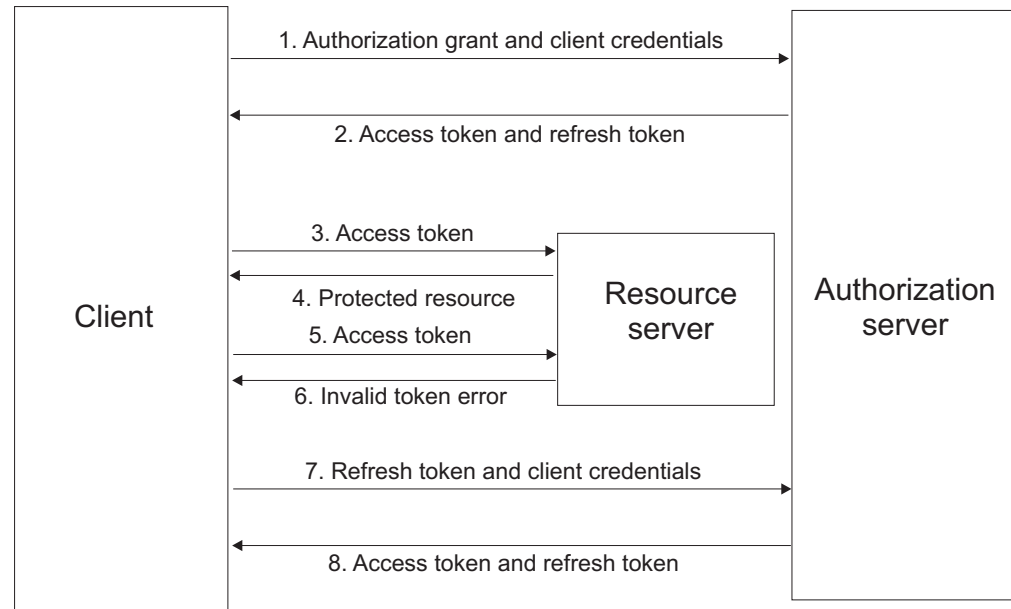


The authorization code workflow diagram involves the following steps:

1. The OAuth client initiates the flow when it directs the user agent of the resource owner to the authorization endpoint. The OAuth client includes its client identifier, requested scope, local state, and a redirection URI. The authorization server sends the user agent back to the redirection URI after access is granted or denied.
2. The authorization server authenticates the resource owner through the user agent and establishes whether the resource owner grants or denies the access request.
3. If the resource owner grants access, the OAuth client uses the redirection URI provided earlier to redirect the user agent back to the OAuth client. The redirection URI includes an authorization code and any local state previously provided by the OAuth client.
4. The OAuth client requests an access token from the authorization server through the token endpoint. The OAuth client authenticates with its client credentials and includes the authorization code received in the previous step. The OAuth client also includes the redirection URI used to obtain the authorization code for verification.
5. The authorization server validates the client credentials and the authorization code. The server also ensures that the redirection URI received matches the URI used to redirect the client in Step 3. If valid, the authorization server responds back with an access token.

The authorization server can be the same server as the resource server or a separate entity. A single authorization server can issue access tokens accepted by multiple resource servers.

Authorization code flow with refresh token



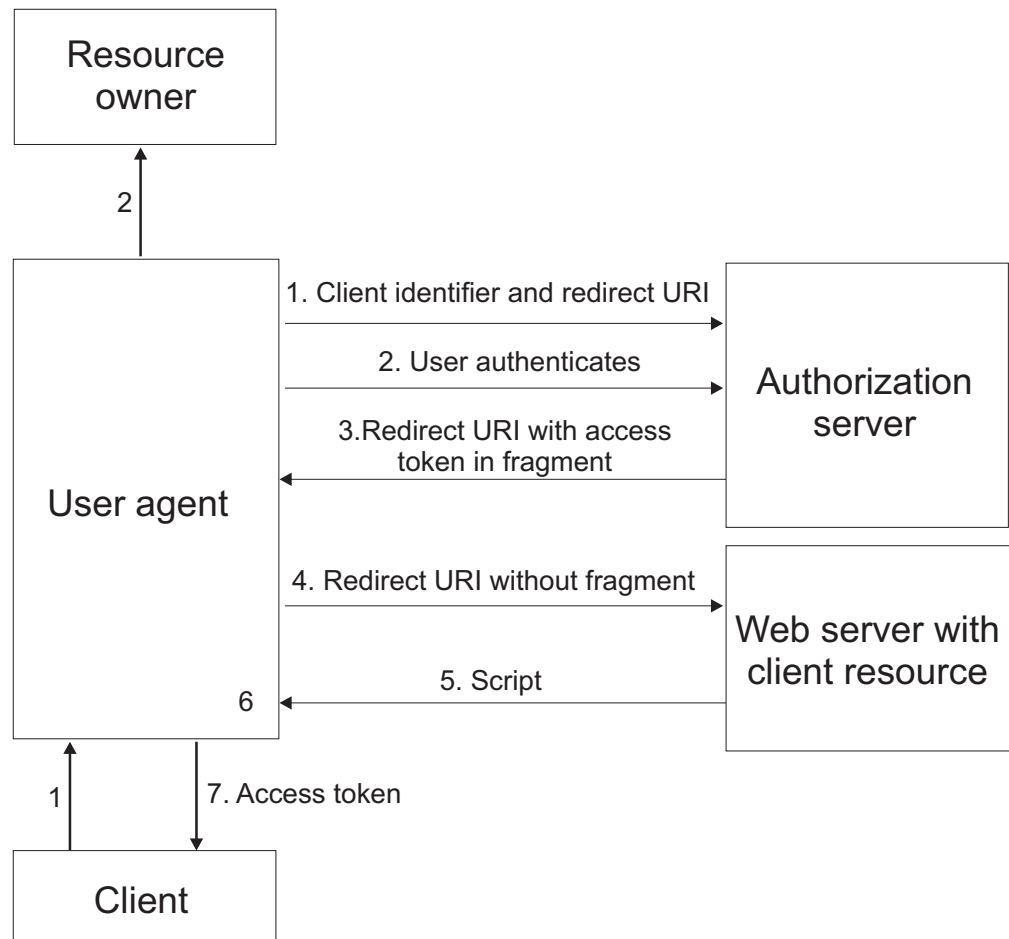
The authorization code workflow with refresh token diagram involves the following steps:

1. The OAuth client requests an access token by authenticating with the authorization server with its client credentials, and presenting an authorization grant.
2. The authorization server validates the client credentials and the authorization grant. If valid, the authorization server issues an access token and a refresh token.
3. The OAuth client makes a protected resource request to the resource server by presenting the access token.
4. The resource server validates the access token. If the access token is valid, the resource owner serves the request.
5. Repeat steps 3 and 4 until the access token expires. If the OAuth client knows that the access token has expired, skip to Step 7. Otherwise, the OAuth client makes another protected resource request.
6. If access token is not valid, the resource server returns an error.
7. The OAuth client requests a new access token by authenticating with the authorization server with its client credentials, and presenting the refresh token.
8. The authorization server validates the client credentials and the refresh token, and if valid, issues a new access token and a new refresh token.

Implicit grant flow

The implicit grant type is suitable for clients that are not capable of maintaining their client credentials confidential for authenticating with the authorization server. An example can be in the form of client applications that are in a user agent, typically implemented in a browser using a scripting language such as JavaScript.

As a redirection-based flow, the OAuth client must be able to interact with the user agent of the resource owner, typically a web browser. The OAuth client must also be able to receive incoming requests through redirection from the authorization server.



The implicit grant workflow diagram involves the following steps:

1. The OAuth client initiates the flow by directing the user agent of the resource owner to the authorization endpoint. The OAuth client includes its client identifier, requested scope, local state, and a redirection URI. The authorization server sends the user agent back to the redirection URI after access is granted or denied.
2. The authorization server authenticates the resource owner through the user agent and establishes whether the resource owner grants or denies the access request.
3. If the resource owner grants access, the authorization server redirects the user agent back to the client using the redirection URI provided earlier. The redirection URI includes the access token in the URI fragment.
4. The user agent follows the redirection instructions by making a request to the web server without the fragment. The user agent retains the fragment information locally.
5. The web server returns a web page, which is typically an HTML document with an embedded script. The web page accesses the full redirection URI

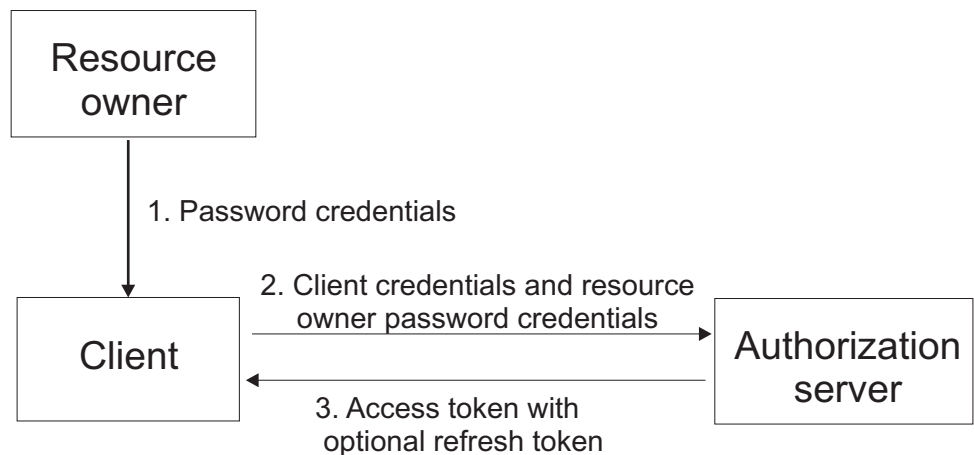
including the fragment retained by the user agent. It can also extract the access token and other parameters contained in the fragment.

6. The user agent runs the script provided by the web server locally, which extracts the access token and passes it to the client.

Resource owner password credentials flow

The resource owner password credentials grant type is suitable in cases where the resource owner has a trust relationship with the client. For example, the resource owner can be a computer operating system of the OAuth client or a highly privileged application.

You can only use this grant type when the OAuth client has obtained the credentials of the resource owner. It is also used to migrate existing clients using direct authentication schemes by converting the stored credentials to an access token.



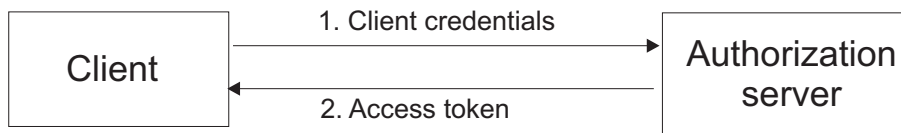
The resource owner password credentials workflow diagram involves the following steps:

1. The resource owner provides the client with its user name and password.
2. The OAuth client requests an access token from the authorization server through the token endpoint. The OAuth client authenticates with its client credentials and includes the credentials received from the resource owner.
3. After the authorization server validates the resource owner credentials and the client credentials, it issues an access token and optionally a refresh token.

Client credentials flow

The client credentials flow is used when the OAuth client requests an access token using only its client credentials. This flow is applicable in one of the following situations:

- The OAuth client is requesting access to the protected resources under its control.
- The OAuth client is requesting access to a different protected resource, where authorization has been previously arranged with the authorization server.



The client credentials workflow diagram involves the following steps:

1. The OAuth client requests an access token from the token endpoint by authenticating with its client credentials.
2. After the authorization server validates the client credentials, it issues an access token.

Client authentication considerations at the OAuth 2.0 token endpoint

The OAuth 2.0 token endpoint is used for direct communications between an OAuth client and the authorization server.

The OAuth 2.0 token endpoint is used for direct communications between an OAuth client and the authorization server. The token endpoint is used to obtain an OAuth token.

The client type, whether public or confidential, determines the authentication requirements of the OAuth 2.0 token endpoint. The ISAM for Mobile runtime is responsible for authenticating the client by using the `client_id` and `client_secret` in sending the request.

State management

The `state_id` parameter in the STSUniversalUser module is used as a key to store or retrieve state information for each invocation of the trust chain of an OAuth flow.

Security Access Manager for Mobile provides sample mapping rules. These sample mapping rules use state management API and are applicable to OAuth 2.0 protocols. You can get the sample mapping rules from the File downloads section.

OAuth 2.0

OAuth 2.0 tokens, such as grants, access tokens, and refresh tokens, have a `state_id` parameter that is used in Security Token Service mapping rules. The `state_id` parameter maintains state between associated Security Token Service calls in an OAuth 2.0 flow.

The OAuth 2.0 mapping rule uses the `state_id` as the key to issue an authorization grant. The key is used to add the token storage time to a cache. The storage time is then retrieved from the cache during a request for a protected resource.

Figure 10 on page 75 shows a section of the sample JavaScript mapping rule for OAuth 2.0.

```

...
var request_type = null;
var grant_type = null;

// The request type - if none available assume 'resource'
temp_attr = stsuu.getContextAttributes().getAttributeValuesByNameAndType("request_type", "urn:ibm:names:ITFIM:oauth:request");
if (temp_attr != null && temp_attr.length > 0) {
    request_type = temp_attr[0];
} else {
    request_type = "resource";
}

// The grant type
temp_attr = stsuu.getContextAttributes().getAttributeValuesByNameAndType("grant_type", "urn:ibm:names:ITFIM:oauth:body:param");
if (temp_attr != null && temp_attr.length > 0) {
    grant_type = temp_attr[0];
}

/* The following demonstrates the use of the state management API.
 *
 * request_type = 'authorization' ==> Store the UTC time of the request into a cache
 *           with state_id as key [authorization_code, implicit]
 * request_type = 'access_token' && grant_type = 'client_credentials' ==> Store the UTC time of the request
 *           into a cache with state_id as key [client_credentials]
 * request_type = 'access_token' && grant_type = 'password' ==> Store the UTC time of the request into a cache
 *           with state_id as key [password]
 * request_type = 'resource' ==> Retrieve the stored time and put it into an attribute named recovered_state
 *
 * It also stores the flow type we are in be used later to detect if this is a client_credentials two-legged flow or not.
 */
if (request_type == "authorization" || (request_type == "access_token" &&
    (grant_type == "client_credentials" || grant_type == "password"))) {
    var curr_utc_time = "State storage time was: " + IDMappingExtUtils.getCurrentTimeStringUTC();
    IDMappingExtUtils.getIDMappingExtCache().put(state_id, curr_utc_time, 1000);
} else if (request_type == "resource") {
    var recovered_state = IDMappingExtUtils.getIDMappingExtCache().get(state_id);

    if (recovered_state != null) {
        var state_arr = java.lang.reflect.Array.newInstance(java.lang.String, 1);
        state_arr[0] = recovered_state;
        stsuu.addContextAttribute(new Attribute("recovered_state",
            "urn:ibm:names:ITFIM:oauth:response:attribute", state_arr));
    }
}
...

```

Figure 10. OAuth 2.0 JavaScript sample code with state management

Trusted clients management

Security Access Manager for Mobile stores trusted client information that is based on the decisions of a resource owner on which clients to trust.

In an OAuth 2.0 flow, the resource owner is asked to provide consent on the scopes that are requested by a client to access the protected resource. The resource owner can either grant permission or deny the client from its access request.

The OAuth server or authorization server uses the trusted clients manager to manage information about trusted clients.

Administrators can configure the behavior of the trusted clients manager in the API protection page. They can configure whether a resource owner is prompted for consent in the Authorization code flow or the Implicit grant flow.

The following configuration options are available:

- Never prompt a resource owner for consent - Resource owners are never prompted for consent and the authorization decision defaults to allow access to the resource.

- Always prompt a resource owner for consent - Resource owners are always prompted for consent even if the client was previously allowed to access the resource.
- Prompt the resource owner once and remember consent - Resource owners are prompted once for consent and later allows access to the resource.

Note: For the Prompt once and remember configuration options, the trusted client manager verifies whether the resource owner previously provided consent on the scopes that are requested by a client.

Configuring API protection

The API protection uses the OAuth 2.0 protocol. To configure the API protection, you must create a definition and a client.

You must then attach the API protection definition to a resource.

Creating API protection definition

Create API protection definitions to configure the settings that dictate the behavior of how resources are accessed.

About this task

The API protection definition page has several sections:

Name Specify a unique name for the API protection definition. This field is mandatory.

Description

Provide a brief description of the API protection definition.

Grant Types

Specify the list of supported authorization grant types. You must select at least one grant type. Authorization code is enabled by default.

- Authorization code
- Resource owner username and password
- Client credentials
- Implicit

Token Management

Specify values for:

- Access token lifetime (seconds)
- Access token length
- Enforce single-use authorization grant
- Authorization code lifetime (seconds) - only configurable if Authorization code grant type is selected.
- Authorization code length - only configurable if Authorization code grant type is selected.
- Issue refresh token - only available for the Authorization code and Resource owner username and password grant types.
 - Maximum authorization grant lifetime (seconds)
 - Refresh token length
 - Enforce single access token per authorization grant
 - Enable PIN policy


- PIN length

Trusted Clients and Consent

Specify the behavior of consent to authorize prompt during client authorization for Authorization code and Implicit grant types.

- Always prompt
- Never prompt
- Prompt once and remember

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **API Protection**.
4. Click **Definitions**.
5. Click .
6. In the **Name** field, type a unique name for the definition.

Note: The name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

7. In the **Description** field, provide a brief description about the definition.
8. Click **Grant Types**.
9. Select the list of supported grant types.
10. Click **Token Management**.
11. Provide the following information:

Access token lifetime (seconds)

Specifies the validity of an access token in seconds.

When this lifetime expires, the client cannot use the current access token to access the protected resource.

Units: seconds

Default value: 3600 seconds

Minimum value: 1 second

Access token length

The length of an access token.

Units: characters

Default value: 20 characters

Minimum value: 1 character

Maximum value: 500 characters

Enforce single-use authorization grant

If enabled, after an access token is validated, all tokens of the authorization grant are revoked.

Default value: disabled

Authorization code lifetime (seconds)

Specifies the validity of the authorization code in seconds.

This option applies only to an authorization code grant type. The authorization server generates an authorization code and issues it to the client. The client uses the authorization code in exchange for an access token.

Units: seconds

Default value: 300 seconds

Minimum value: 1 second

Authorization code length

The length of an authorization code.

Units: characters

Default value: 30 characters

Minimum value: 1 character

Maximum value: 500 characters

Issue refresh token

This option is only applicable to the Authorization code and Resource owner password credentials grant types. Specifies whether a refresh token is issued to the client. A refresh token is used to obtain a new pair of access token and refresh token.

Maximum authorization grant lifetime (seconds)

Specifies the maximum duration of a grant where the resource owner authorized the client to access the protected resource.

This option is available only if you enable the issuing of a refresh token.

The value for this lifetime must be greater than the values specified for the authorization code and access token lifetimes.

When this lifetime expires, the resource owner must reauthorize the client to obtain an authorization grant to access the protected resource.

Units: seconds

Default value: 604800 seconds

Minimum value: 1 second

Refresh token length

This option is available only if you enable the issuing of refresh tokens. It defines the length of a refresh token.

Units: characters

Default value: 40 characters

Minimum value: 1 characters

Maximum value: 500 characters

Enforce single access token per authorization grant

This option is available only if you enable the issuing of refresh tokens. If enabled, previously granted access tokens are revoked after a new access token is generated by presenting the refresh token to the authorization server.

Default value: enabled

Enable PIN policy

This option is only available if you enable the issuing of refresh tokens. Enabling the PIN policy is designed to provide more protection during the exchange of a refresh token for a new pair of access token and refresh token. When enabled, you must configure the PIN length.

PIN Length

Defines the length of a PIN. This option is enabled only when PIN is enabled. You can use the `runtime.hashAlgorithm` runtime parameter to configure the algorithm that is used to hash the PIN before it is stored. For more information, see “Advanced configuration properties” on page 108.

Units: characters

Default value: 3 characters

Minimum value: 12 characters

12. Click **Trusted Clients and Consent**.

13. Specify the following information:

Always prompt

The user is always prompted to provide their consent for a new authorization grant. The consent decision is not stored.

Never prompt

The consent to authorize prompt is never shown to the user. The consent is implicit and is not stored.

Prompt once and remember

The user is prompted for consent to authorize when a previous consent for the client with the particular scope is not already stored. If consent is granted, the decision is stored by the Trusted Client Manager.

14. Click **Save**.

What to do next

- You must attach definitions to resources and publish the attachments for definitions to be implemented. See “Managing policy attachments” on page 83.
- Register an API protection client.
- You must deploy the pending changes. See, Chapter 13, “Deploying pending changes,” on page 119

Managing API protection definitions

An API protection definition is a set of configurations that define how resources are accessed.

About this task

You can add, modify, and delete definitions.

Procedure


1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **API Protection**.

4. Click **Definitions**.
5. Perform one or more of the following actions:

Add definitions

Click . See [Creating an API protection definition for details](#).

Modify definitions


- a. Select a definition in the list of definitions.
- b. Click .
- c. Complete the properties for the definition.

Note:

You cannot modify the definition name and grant types. See [Creating an API protection definition for details](#).

- d. Click **Save**.

Delete definitions

- a. Select a definition or press Ctrl and select multiple definitions in the definition list.
- b. Click . Confirm the deletion. Click **OK** to continue or click **Cancel**.

Note: A definition cannot be deleted if there are client associated with it or it is attached to a resource.

6. Click **Save**.
7. When you add, modify or delete a definition, a message indicates that there are changes to deploy. If you are finished with the changes, deploy them.
For more information, see Chapter 13, “Deploying pending changes,” on page 119.

PIN policy

IBM Security Access Manager for Mobile extends OAuth 2.0 capabilities with a PIN policy.

The PIN policy provides the capability of protecting a refresh token with a PIN provided by the API protection client. An administrator can configure the API protection definition to enable the PIN policy for the grant types that issue a refresh token. The two grant types that issue a refresh token are Authorization code and the Resource owner password credentials.


When enabled, the client is required to send a PIN as a parameter in the first access token request. The parameter name is **pin**. The parameter value consists of digits of the length that is configured in the API protection definition. The client must submit the same PIN on subsequent requests when exchanging a refresh token for a new access token.

The PIN policy can be configured to use various hash algorithms to hash and store the PIN. Use the **runtime.hashAlgorithm** configuration parameter to specify the hash algorithm. For more information about configuring the hash algorithm, see *Runtime properties* in “Advanced configuration properties” on page 108

Registering an API protection client

Register OAuth API protection clients in the Clients panel. Clients are the entities against which OAuth access and refresh tokens are granted at runtime.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **API Protection**.
4. Click **Clients**.
5. Click  .
6. Specify the following information:

Client name

Specify a meaningful client identifier for each client registration. You can use this value to search for client registrations.

API definition

Specifies the related Definition, which owns and defines the client. A Definition can own many client registrations but a client registration can belong to only one Definition. When you create a client, a list of available Definitions are available. When a client is created, this value cannot be modified.

Confidential

Specify whether the client type is confidential. A confidential client type requires a client secret. Enable this feature if you want the client to require a client secret.

Client secret

This field is enabled only if the client is indicated as confidential. Specify a client secret that is used to authenticate an OAuth client at runtime. It is mandatory for all clients that belong to API protection definitions where the client type is Confidential and the client credentials grant type is enabled. Click **Generate** to have a client secret that is generated for you or specify your own secret.

Redirect URI (Optional)

Specify the redirect URI to use for the client.

Company name

Specify the name of the company for this client.

Contact name (Optional)

Specify a name of the contact person for this client.

Email address (Optional)

Specify the email address of the contact person for this client.

Telephone number (Optional)

Specify the telephone number of the contact person for this client.

Contact type (Optional)

Select the contact type from the list:

- Administrative
- Support
- Technical
- Billing

- Other

Other information (Optional)

Specify extra information about the client contact.

7. Click **OK**.

Managing registered API protection clients

Manage registered OAuth API protection clients.

About this task

You can search and delete clients. You can search for API protection clients based on the following values:

- Client name
- Client ID
- API protection definition



Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **API Protection**.
4. Click **Clients**.
5. Perform one or more of the following actions:

View and filter clients


You can filter for client name, client ID, and API protection definition.

Take any of the following actions to filter your view:


- Select the  **Details View** to view client name, client ID, and API protection definition.
- Select the  **List View** to view only the name of the client.
- Type a term, such as an client name, client ID, and API protection definition in the **Filter** field to list clients that use that term. Any part of the values for client name, client ID, or API protection definition that match is applied by the filter and is displayed in the search results. Click x to clear the **Filter** field.
- Sort the client list by column with the up or down arrow on each column. For example, you can view the list of clients that are sorted by the **Clients** column in ascending order by clicking the up arrow.

Modify clients

Attention: Ensure that the modification does not affect a current policy or configuration. If you modify a client that is in-use, the policy or configuration that uses the client might stop working.

- a. Select the client you want to modify.
- b. Click .
- c. Complete the properties for the clients.
- d. Click **OK**.

Delete clients

- a. Select a client or press and hold the Ctrl key and select multiple clients to remove.
- b. Click . Confirm the deletion. Click **OK** to continue or click **Cancel**.

When you delete a client:

- The client registration is removed from the database.
 - All tokens issued against that client is removed.
6. When you add, modify or delete a client, a message indicates that there are changes to deploy. If you are finished with the changes, deploy them.
For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Managing policy attachments

Attach policies or API protection definitions to resources so that the policies and definitions can be enforced.

Before you begin

You must create policies, policy sets, or API protection definitions.

When you create policies, policy sets, or API protection definitions you cannot use them until you publish them to resources. Once policies, policy sets, or API protection definitions are published, they are enforced during the evaluation of access requests.

Since policies and API protection definitions are attached to resources, you must first configure resources. This means that policy and API protection definition attachment is dependent on prior installation and configuration of a IBM Security Access Manager for Web runtime component and reverse proxy server. These prior configuration tasks include the creation of a Security Access Manager for Web protected object space. In a Web environment, this configuration includes creation of a junction and resources (objects) that are protected by that junction.

About this task

You can:

- Add a resource
- Add a policy or API protection definition attachment to a resource
- Remove a policy or API protection definition attachment from a resource
- Delete a resource
- Publish a policy or API protection definition attachment


When a deployment is fully configured, the Resources panel displays three levels of entries. The top-level entry is the web container that contains the protected object space for a server instance. The second level shows the resources in the protected object space. The third level lists the policies and API protection definitions that are attached to each resource.

Tip: The user interface provides a quick filter feature for use on the top-level entry. Use the quick filter to search for a specific top-level entry. Enter the first few characters of the web container, and the list displays only the entries that contain the specified characters.


Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. Click **Resources**.
5. Perform one or more of the following actions:

Add a resource

- a. Click .
- b. In the **Web Container** field, click the down arrow icon to display a list of web containers. Select an entry.
For example, the list of web containers is the WebSEAL protected object space that is defined directly under /WebSEAL.
- c. Specify a resource. You can either enter the name of the resource or browse for it:
 - If you know the name of the resource within the protected object space, you can enter it in the **Resource** field.
For example, /myserver-jct/benefits/medical
 - If you select **Browse**, you can select a resource from the tree hierarchy. Expand the hierarchy to see all available resources.
The display of the hierarchy is based on the structure of the WebSEAL protected object space:
 - In some cases, not all resources are displayed because the WebSEAL protected object space is a sparse tree. For example, you might see only the resource /myserver-jct/benefits. You can select this resource and click **OK** to add it to the **Resource**. You can then add /myserver-jct/benefits/medical.
 - In some cases, you cannot view the object space for the web server junction. For example, if the administrator did not install the IBM Security Access Manager **querycontents** script on the application server, you cannot see the junction contents. In these cases, you can enter the resource path manually.
- d. Click **Save**.
- e. Next, attach a policy to the resource.

Attach a policy or API protection definition to a resource

- a. Select a resource node and click  **Attach**.
- b. In the Attach Policies panel, select **Policies** or **Policy Sets** or **API Protection**.
- c. From the displayed list, select one or more policies or policy sets or API protection definitions.

Tip: You can type the name of the applicable policy or policy set or API protection definition in the quick filter.


Notes:

- You can attach both individual policies, policy sets, or API protection definitions.
- You cannot attach policies or policy sets to a resource where that resource already has API protection definitions attached.

- You cannot attach API protection definitions to a resource where that resource already has policies and policy sets attached.
- d. Click **OK** to save your changes.


Note: The policy or API protection definition remains inactive until you publish it.

Remove a policy or API protection definition attachment

- To remove a policy or API protection definition attachment from a resource, select the policy node and click .
- When prompted, confirm the deletion.

Note: You must publish the change.


Delete a resource

- To delete a resource and all attached policies or API protection definitions, select the resource node and click .
- When prompted, confirm the deletion.

When you delete a resource:

- You cannot delete the server node.
- You do not have to manually publish the change. The deletion is automatically published.


Publish a policy or API protection definition

Select a resource in the resource hierarchy and click  Publish. When the publication completes, the status column for the resource indicates the status and time of the publication.

Note: Activation of the published policy or API protection definition could take up to a minute to complete.

Modify Resource

Note: You can only use this function if policy or policy sets are attached to the given resource.

- Select a resource node and click .
- In the Modify Resource panel, you can modify the **Policy Combining Algorithm**. Choose the preferred algorithm:
 - **Deny access if any attached policy returns deny**
This algorithm means that if multiple policies or API protection definitions are attached to a resource, and any one of those policies or API protection definitions returns Deny, then the access request is denied.
 - **Permit access if any attached policy returns permit**
This algorithm means that if multiple policies or API protection definitions are attached to a resource, and any one of those policies or API protection definitions returns Permit, then the access request is permitted.

Uploading OAuth response files

Use the local management interface to upload your own custom OAuth response files.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings > Manage > Reverse Proxy**.
2. Select a reverse proxy.
3. Select **Manage > Management Root**.
4. Select the **oauth** folder.
5. Select **Manage > Import**.
6. Click **Browse**.
7. Browse to the file you want to import.
8. Click **Open**.
9. Click **Import**.


Managing OAuth 2.0 mapping rules

The OAuth 2.0 mapping rules are JavaScript code that run during the OAuth 2.0 flow. Use the rules to customize the methods to use for the OAuth 2.0 flow. You can view, export, and replace OAuth mapping rules.


Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **API Protection**.
4. Click **Advanced**.
5. Perform one or more of the following actions:

View a mapping rule


- a. Select a mapping rule.
- b. Click . The View Mapping Rule panel opens. The content of the mapping rule is displayed.
- c. Click **OK** to close the panel.

Export a mapping rule

- a. Select a mapping rule.
- b. Click .
- c. Choose a location and save the file.

Replace a mapping rule:

Note: Use an existing mapping rule as the basis for the updated mapping rule.

- a. Select a mapping rule that you want to replace.
- b. Click . The Replace Mapping Rule panel opens.
- c. Click the field or **Browse** and select a file.
- d. Click **OK** to upload the mapping rule.

6. When you replace a mapping rule, the appliance displays a message that there are undeployed changes. If you are finished making the changes, deploy them. For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Related reference:

“OAuth 2.0 mapping rule methods”

Methods are available for you to use in the `PreTokenGeneration` and `PostTokenGeneration` mapping rules.

OAuth 2.0 mapping rule methods

Methods are available for you to use in the `PreTokenGeneration` and `PostTokenGeneration` mapping rules.

Sample mapping rules are in **Manage System Settings > Secure Settings > File Downloads > mga > example > demo > demo_rules**.

The following limitations affect the attribute keys and values that are associated with the `state_id` by using the `OAuthMappingExtUtils` class:

- Keys cannot be null or empty.
- Values cannot be null but can be empty.
- Key-value pairs that are associated are read and write-allowed and not-sensitive.
- Some keys are reserved for system use and cannot be modified by this utility. For example, the keys and values for the API PIN protection.

associate

```
public static boolean associate(  
    String stateID,  
    String attrKey,  
    String attrValue  
)
```

This method associates the attribute key-value pair to the authorization grant state ID. Use the following parameters:

- `stateID` - The state ID of the authorization grant. This parameter cannot be null or empty.
- `attrKey` - The attribute key. This parameter cannot be null or empty. The maximum length is 256 characters.
- `attrValue` - The attribute value. This parameter cannot be null. The maximum length is 256 characters.

These responses come from the runtime after association.

- True if successful.
- False if not successful.

disassociate

```
public static String disassociate(  
    String stateID,  
    String attrKey  
)
```

This method disassociates the attribute key-value pair from the authorization grant state ID. Use the following parameters:

- `stateID` - The state ID of the authorization grant. This parameter cannot be null or empty.

- attrKey - The attribute key to disassociate. This parameter cannot be null or empty.

These responses come from the runtime after disassociation.

- The previous string value that is associated with the state ID and attribute key if successful.
- Null if not successful. For example, when association is never made.

getAssociationKeys

```
public static String[] getAssociationKeys(
    String stateID
)
```

This method gets all the attribute keys that are associated with the authorization grant state ID. Use the following parameter:

- stateID - The state ID of the authorization grant. This parameter cannot be null or empty.

These responses come from the runtime after it gets the association keys.

- A string array of all attribute keys that are associated with the authorization grant state ID if successful.
- Null if state ID is wrong, if there are problems during retrieval from the token cache, or there are no associated attributes.

getAssociation

```
public static String getAssociation(
    String stateID,
    String attrKey
)
```

This method gets the attribute value from the authorization grant state ID and attribute key. Use the following parameters:

- stateID - The state ID of the authorization grant. This parameter cannot be null or empty.
- attrKey - The attribute key value that you want to be returned. This parameter cannot be null or empty.

These responses come from the runtime after it gets the association keys.

- The string value that is associated with the state ID and attribute key if successful.
- Null if not successful. For example, when the association is never made.

throwSTSException

```
public static void throwSTSException(
    String message
)
```

This method throws an STSException. Use the following parameter:

- message - The message to be printed along with the stack trace.

throwSTSUserMessageException

```
public static void throwSTSUserMessageException(
    String message
)
```

This method throws an STSUserMessageException. Use the following parameter:

- message - The message to be printed along with the stack trace.

HTTP GET

```
public static HttpResponse httpGet(  
    String url,  
)
```

This method sends a HTTP GET request. Use the following parameters:

- `url` - The URL to send the GET request.

This request returns an `HttpResponse`. The request becomes null when there is no response or when it contains wrong parameters.

HTTP GET

```
public static HttpResponse httpGet(  
    String url,  
    Map headers,  
    String httpsTrustStore,  
    String basicAuthUsername,  
    String basicAuthPassword,  
    String clientKeyStore,  
    String clientKeyAlias  
)
```

This method sends a HTTP GET request. Use the following parameters:

- `url` - The URL to send the GET request.
- `headers` - A map of the headers.
- `httpsTrustStore` - The truststore name. If this parameter is null and SSL connection is required, use the default truststore.
- `basicAuthUsername` - The user name for basic authentication. If this parameter is null, the basic authentication is not enabled.
- `basicAuthPassword` - The password that is used for basic authentication. If null, basic authentication is not enabled.
- `clientKeyStore` - The client keystore name. If this parameter is null, client certificate authentication is not enabled.
- `clientKeyAlias` - The client key alias. If this parameter is null, client certificate authentication is not enabled.

This request returns an `HttpResponse`. The request becomes null when there is no response or when it contains wrong parameters.

HTTP POST

```
public static HttpResponse httpPost(  
    String url,  
    Map params  
)
```

This method sends a HTTP POST request. Use the following parameters:

- `url` - The URL to send the POST request.
- `params` - A map of the parameters.

This request returns an `HttpResponse`. The request becomes null when there is no response or when it contains wrong parameters.

HTTP POST

```
public static HttpResponse httpPost(  
    String url,  
    Map headers,  
    Map params,  
    String httpsTrustStore,  
    String basicAuthUsername,  
)
```

```

        String basicAuthPassword,
        String clientKeyStore,
        String clientKeyAlias
    )

```

This method sends a HTTP POST request. Use the following parameters:

- `url` - The URL to send the GET request.
- `headers` - A map of the headers.
- `httpsTrustStore` - The truststore name. If this parameter is null and SSL connection is required, use the default truststore.
- `basicAuthUsername` - The user name for basic authentication. If this parameter is null, the basic authentication is not enabled.
- `basicAuthPassword` - The password that is used for basic authentication. If null, basic authentication is not enabled.
- `clientKeyStore` - The client keystore name. If this parameter is null, client certificate authentication is not enabled.
- `clientKeyAlias` - The client key alias. If this parameter is null, client certificate authentication is not enabled.

This request returns an `HttpResponse`. The request becomes null when there is no response or when it contains wrong parameters.

HttpResponse

```
public int getCode()
```

This method gets the HTTP response code.

This method returns the `HttpResponse` code. The method returns -1 if the response code is not set.

HttpResponse

```
public String getBody()
```

This method gets the HTTP response body.

This method returns the HTTP response body. The method returns an empty string if the response body is not set.

HttpResponse

```
public String[] getHeaderKeys()
```

This method gets the HTTP response header keys.

This method returns a string array of the HTTP response header keys. This method returns an empty array if there are no headers.

HttpResponse

```
public String[] getHeaderValues(
    String key
)
```

This method gets the HTTP response header values for the key.

This method returns a string array of the HTTP response header values for the key. This method returns an empty array if there is no value for the header key.

For more information, see the Javadoc in **Manage System Settings > Secure Settings > File Downloads > mga > doc**.

OAuth 2.0 template pages

This topic contains references about the HTML template pages.

OAuth 2.0 template page for consent to authorize

The authorization server uses this page to determine and store user consent information about which OAuth clients are authorized to access the protected resource. This page also indicates scopes that the OAuth client requests.

The Security Access Manager for Mobile provides an HTML page template called `user_consent.html`. The macros in the template are specifically for an OAuth 2.0 flow.

Security Access Manager for Mobile stores the decisions made by the resource owner about which OAuth clients to trust. The resource owner is not prompted every time the same OAuth client requests authorization to access the protected resource.

The authorization request from the OAuth client shows a list of approved scopes, and a list of scopes to be approved. These lists are shown in the consent page and can be of indeterminate length. The template supports multiple copies of stanzas that are repeated once for each scope in either list.

This template file provides several replacement macros:

@OAUTH_AUTHORIZE_URI@

This macro is replaced with the URI for the authorization endpoint.

@OAUTH_CLIENT_COMPANY_NAME@

This macro is replaced with the name of the company that is requesting access the protected resource.

@CLIENT_ID@

This macro is replaced with the `client_id` parameter specified in the authorization request.

@REDIRECT_URI@

This macro is replaced with the redirect URI that the authorization server uses to send the authorization code to. The value depends on the following items:

- Redirect URI that is entered during partner registration
- `oauth_redirect` parameter specified in the authorization request

@STATE@

This macro is replaced with the state parameter specified in the authorization request.

@RESPONSE_TYPE@

This macro is replaced with the `response_type` parameter specified in the authorization request.

@OAUTH_CUSTOM_MACRO@

This macro is replaced with trusted client information that contains additional information about an authorized OAuth client.

@USERNAME@

This macro is replaced with the Security Access Manager for Mobile user name.

@OAUTH_OTHER_PARAM_REPEAT@

A multi-valued macro that belongs inside a [RPT oAuthOtherParamsRepeatable] repeatable replacement list. The values show the list of extra parameter names.

@OAUTH_OTHER_PARAM_VALUE_REPEAT@

A multi-valued macro that belongs inside a [RPT oAuthOtherParamsRepeatable] repeatable replacement list. The values show the list of extra parameter values.

@OAUTH_TOKEN_SCOPE_REPEAT@

A multi-valued macro that belongs either inside [RPT oAuthTokenScopePreapprovedRepeatable] or [RPT oAuthTokenScopeNewApprovalRepeatable] repeatable replacement lists. The values inside the [RPT oAuthTokenScopePreapprovedRepeatable] show the list of token scopes that have been previously approved by the resource owner. Alternatively, the values inside the [RPT oAuthTokenScopeNewApprovalRepeatable] show the list of token scopes that have *not* yet been approved by the resource owner.

@CONSENT_FORM_VERIFIER@

This macro is replaced with a unique identifier for the consent_form_verifier parameter value. The consent_form_verifier parameter value is automatically generated by the authorization server. The parameter name and value must not be modified.


```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>OAuth 2.0 - Consent to Authorize</title>
    <style type="text/css">
      @import "/sps/static/styles.css";
    </style>
  </head>
  <body>
    <div class="header">
      <div class="brandingLogo"></div>
    </div>
    <div class="content">
      <div class="contentHeader">
        <h1 class="pageTitle">OAuth 2.0 - Consent to Authorize</h1>
        <div class="instructions"></div>
      </div>
      <div class="pageContent">
        <form action="@OAUTH_AUTHORIZE_URI@" method="post">
          <p>The following site is requesting access to an OAuth 2.0 protected resource:</p>
          <div class="sectionTitle">
            <p><b>@OAUTH_CLIENT_COMPANY_NAME@</b></p>
          </div>
          <p>The client type is: @CLIENT_TYPE@</p>
          <br/>
          <p>The client provided the following OAuth 2.0 request parameters:</p>
          <br/>
          <ul style="margin-left: 20px">
            <li>Client Id: @CLIENT_ID@</li>
            <li>Redirect URI: @REDIRECT_URI@</li>
            <li>State: @STATE@</li>
            <li>Response Type: @RESPONSE_TYPE@</li>
          </ul>
          <br/>
          <p>By approving this request you will be providing delegated authorization on behalf of:</p>
          <p><b>@USERNAME@</b></p>
          <br/>
          <p>The client provided the following extra request parameters:</p>
          <!-- START NON-TRANSLATABLE -->
          <ul style="margin-left: 20px">
            [RPT oauthOtherParamsRepeatable]
            <li>@OAUTH_OTHER_PARAM_REPEAT@=@OAUTH_OTHER_PARAM_VALUE_REPEAT@</li>
            <input type="hidden" name="@OAUTH_OTHER_PARAM_REPEAT@"
            value="@OAUTH_OTHER_PARAM_VALUE_REPEAT@" />
            [ERPT oauthOtherParamsRepeatable]
          </ul>
          <!-- END NON-TRANSLATABLE -->
          <br/>
          <p>The client requested the following token scopes that have been previously approved:</p>
          <!-- START NON-TRANSLATABLE -->
          <ul style="margin-left: 20px">
            [RPT oauthTokenScopePreapprovedRepeatable]
            <li>@OAUTH_TOKEN_SCOPE_REPEAT@</li>
            <input type="hidden" name="scope" value="@OAUTH_TOKEN_SCOPE_REPEAT@" />
            [ERPT oauthTokenScopePreapprovedRepeatable]
          </ul>
          <!-- END NON-TRANSLATABLE -->
          <br/>
          <p>The client requested the following token scopes that have not yet been approved:</p>
          <!-- START NON-TRANSLATABLE -->
          [RPT oauthTokenScopeNewApprovalRepeatable]
          <input type="checkbox" name="scope" value="@OAUTH_TOKEN_SCOPE_REPEAT@"
          checked="checked" /><label>@OAUTH_TOKEN_SCOPE_REPEAT@</label><br />
          [ERPT oauthTokenScopeNewApprovalRepeatable]
          <!-- END NON-TRANSLATABLE -->
          <p/>
          <br />
          <p>Would you like to approve access to this scope?</p>
          <br/>
          <input type="hidden" name="consent_form_verifier" value="@CONSENT_FORM_VERIFIER@" />
          <!--
            The scope parameters can be:
            1. Requested as part of the redirect for authorization by the client
               by appending them to the authorize URL as query string parameters, and/or
            2. If not requested by the client, and you know what authorization options
               are valid for the protected resources being requested, you may
               also manually prompt for them in this page template as demonstrated
               by the following example scope's
          -->
          <!--
            <table>
              <tr>
                <td>Scopes to be authorized:&nbsp;</td>
                <td>Scope 1</td><input type="checkbox" name="scope" value="token_scope_1" /></td>
                <td>Scope 2</td><input type="checkbox" name="scope" value="token_scope_2" /></td>
                <td>Scope 3</td><input type="checkbox" name="scope" value="token_scope_3" /></td>
              </tr>
            </table>
          -->
          <table>
            <tr>
              <td>Permit&nbsp;</td>
              <td><input type="radio" name="trust_level" value="permit" checked /></td>
            </tr>
            <tr>
              <td>Deny&nbsp;</td>
              <td><input type="radio" name="trust_level" value="deny" /></td>
            </tr>
          </table>
          <br />
          <div class="controls">
            <input class="submitButton" type="submit" name="submit" value="Submit" style="width: 80px" />
          </div>
        </form>
      </div>
    </div>
  </body>
</html>

```

Figure 11. Template for user_consent.html

OAuth 2.0 template page for errors

Security Access Manager for Mobile uses a generic error template page to show detailed text information when an error occurs in an OAuth 2.0 flow.

The template page is `user_error.html`.

The following replacement macro is supported:

@ERROR_CODE@

This macro is replaced with characters that uniquely identify the error.

@ERROR_DESCRIPTION@

This macro is replaced with the native language support (NLS) text of the error message associated with the error.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>OAuth 2.0 - Error</title>
    <style type="text/css">
      @import "/sps/static/styles.css";
    </style>
  </head>
  <body>
    <div class="header">
      <div class="brandingLogo"></div>
    </div>

    <div class="content">
      <div class="contentHeader">
        <h1 class="pageTitle">OAuth 2.0 - Error</h1>
      <div class="instructions"></div>
    </div>

    <div class="pageContent">
      <p>The following error was encountered while processing
      your OAuth request:</p><br/>

      <p>Error Code: <b>@ERROR_CODE@</b></p><br/>
      <p>Error Description: <b>@ERROR_DESCRIPTION@</b></p><br/>
    </div>
  </body>
</html>
```

Figure 12. HTML template for `user_error`

OAuth 2.0 template page for response

Use this HTML page to show the authorization code of an OAuth client that did not specify a client redirection URI upon partner registration.

When the OAuth client does not specify a client redirection URI or cannot receive redirects, the authorization server does not know where to send the resource owner after authorization. The OAuth client does not receive the authorization code required to exchange for an access token or refresh token.

The Security Access Manager for Mobile provides an HTML template page called `user_response.html`. This page shows the authorization code that the resource owner can provide to a trusted OAuth client.

The following replacement macro is supported:

@OAUTH_CODE@

This macro is replaced with the `oauth_code` parameter specified in authorization response.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>OAuth - Response</title>
    <style type="text/css">
      @import "/sps/static/styles.css";
    </style>
  </head>
  <body>
    <div class="header">
      <div class="brandingLogo"></div>
    </div>
    <div class="content">
      <div class="contentHeader">
        <h1 class="pageTitle">OAuth - Response</h1>
        <div class="instructions"></div>
      </div>
      <div class="pageContent">
        <br />
        <p>Your OAuth client did not provide a redirect URI.
        Supply this value to your client:</p>
        <br />
        <p>OAuth Authorization Code: <span class="client">@OAUTH_CODE@</span></p>
      </div>
    </div>
  </body>
</html>
```

Figure 13. Template for `user_response.html`

OAuth 2.0 template pages for trusted clients management

Security Access Manager for Mobile provides an HTML page template which resource owners can use to show and manage trusted clients information.

There are different trusted clients management template pages for each OAuth protocol. These pages look the same, and use the same replacement macros. The template pages for OAuth 2.0 are named as `clients_manager.html`.

The resource owner establishes the OAuth clients through the `user_consent.html` page during authorization requests.

The templates have the following replacement macros:

@USERNAME@

This macro is replaced with the Security Access Manager for Mobile user name.

@OAUTH_CLIENT_COMPANY_NAME@

A multi-valued macro that belongs inside a `[RPT trustedClients]`

repeatable replacement list. The values are replaced with the name of the company that requests access to the protected resource.

@PERMITTED_SCOPES@

A multi-valued macro that belongs inside a [RPT trustedClients] repeatable replacement list. The values are replaced with the token scopes to which the OAuth client has access.

@OAUTH_CUSTOM_MACRO@

A multi-valued macro that belongs inside a [RPT trustedClients] repeatable replacement list. The values are replaced with trusted client information that contains additional information about an authorized OAuth client.

@OAUTH_CLIENTMANAGERURL@

A multi-valued macro that belongs inside a [RPT trustedClients] repeatable replacement list. The values are replaced with the endpoint of the trusted clients manager.

@UNIQUE_ID@

A multi-valued macro that belongs inside a [RPT trustedClients] repeatable replacement list. The values are replaced with a unique identifier that identifies the trusted clients information for each entry in the list.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8"/>
    <title>OAuth 2.0 Client Manager</title>
    <style type="text/css">
      @import "/sps/static/styles.css";
    </style>
  </head>
  <body>
    <div class="header">
      <div class="brandingLogo"></div>
    </div>
    <div class="content">
      <div class="contentHeader">
        <h1 class="pageTitle">OAuth 2.0 Trusted Clients Manager</h1>
        <div class="instructions"></div>
      </div>
      <div class="pageContent">
        Username: <b>@USERNAME@</b>
        <p />
        <br />
        <p>Trusted Clients</p><br />
        <table class="dataTable">
          <tr class="headerRow">
            <td>Client</td>
            <td>Permitted Scopes</td>
            <td>Additional Information</td>
            <td>Action</td></tr>
<!-- START NON-TRANSLATABLE -->
  [RPT trustedClients]
<!-- END NON-TRANSLATABLE -->
          <tr>
            <td>@OAUTH_CLIENT_COMPANY_NAME@</td>
            <td>@PERMITTED_SCOPES@</td>
            <td>@OAUTH_CUSTOM_MACRO@</td>
            <td><a href="@OAUTH_CLIENTMANAGERURL?action=remove&id=@UNIQUE_ID@"
            class="controls">Remove</a></td>
          </tr>
<!-- START NON-TRANSLATABLE -->
  [ERPT trustedClients]
<!-- END NON-TRANSLATABLE -->
        </table>
      </div>
    </div>
  </body>
</html>

```

Figure 14. Template for `clients_manager.html`

Error responses

An HTTP response indicates the type of error that has occurred when an action in an authorization process fails. The error responses described here are only applicable to Policy Enforcement Point (PEP) error responses.

For more information about OAuth 2.0 error responses for other endpoints, see the OAuth website: <http://www.oauth.net>.

In some circumstances, the following HTTP error responses must be returned to the client:

- 400 Bad Request
- 401 Unauthorized
- 502 Bad Gateway

For the 401 response, an additional WWW-Authenticate header is added to the response in the following format:

```
WWW-Authenticate: OAuth realm = <realm-name>
```

The HTML component of the responses is preinstalled from files that have been specified in the EAS configuration.

See the WebSEAL Administration Guide for details on how to configure response template files for the OAuth EAS.

User self-administration tasks for OAuth

Administrators can configure OAuth to enable users to perform certain self-management tasks.

A common user task is to manage authorization grants. For example, users can view the attributes of an authorization grant. A user can also enable an authorization grant.

Managing OAuth 2.0 authorization grants

You can view your authorization grants and the tokens and attributes of each authorization grant.

About this task

You can complete the following tasks:

- View a list of your OAuth 2.0 authorization grants.
- View the OAuth 2.0 tokens and attributes of an authorization grant.
- Remove an OAuth 2.0 authorization grant.
- Enable an OAuth 2.0 authorization grant.
- Disable an OAuth 2.0 authorization grant.

Procedure

Take one of the following actions:

View your OAuth 2.0 authorization grants and the tokens and attributes of each authorization grant

1. Log in to http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html.
2. Click the ID from the table to view the tokens and attributes of that authorization grant.

Note: You can also use the following URL to go directly to the tokens and attributes of a specific authorization grant: http://hostname/mga/sps/mga/user/mgmt/html/device/grant_attributes.html?id=x.

The query string, `id=x`, indicates the authorization grant that you are trying to access. The `x` represents the ID of the authorization grant.

Remove an OAuth 2.0 authorization grant

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html`.
2. Click **Remove** next to the authorization grant that you want to remove.

Enable an OAuth 2.0 authorization grant

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html`.
2. Select the **Enabled** box next to the authorization grant that you want to enable.

Disable an OAuth 2.0 authorization grant

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html`.
2. Clear the **Enabled** box next to the authorization grant that you want to disable.

Note: Authorization grants can be enabled, disabled, or removed in the authorization grant attribute page too.

“REST services for OAuth 2.0 authorization grant management”

You can use the REST services capability to manage your authorization grants.

REST services for OAuth 2.0 authorization grant management

You can use the REST services capability to manage your authorization grants.

Note: The user must authenticate to use the REST services capability.

REST services usage scenarios

Depending on your usage scenario, type the following URLs into the web page that calls the REST services:

Table 7. REST services instructions

| Method | URL | Request | Response | Response type |
|--------|---|---|---|------------------|
| GET | <code>https://hostname/mga/sps/mga/user/mgmt/grant</code> | There is no JSON payload included in the request. | <pre> :"id","isEnabled": {true false},"clientId":" clientId","tokens": [{"tokenId":"tokenId ","tokenString":"token string",...}] </pre> <p>If the request completes successfully, the HTTP response code is 200.</p> <p>If the request does not complete successfully, the HTTP response is 500.</p> | application/json |

Table 7. REST services instructions (continued)

| Method | URL | Request | Response | Response type |
|--------|---|---|--|------------------|
| GET | https://hostname/mga/sps/mga/user/mgmt/grant/{id} | There is no JSON payload included in the request. | <pre>{"id": "id", "isEnabled": {true false}, "clientId": "clientId", "tokens": [{"tokenId": "tokenId", "tokenString": "tokenString", ...}]}</pre> <p>If the request completes successfully, the HTTP response code is 200.</p> <p>If the request does not complete successfully, the HTTP response is 500.</p> | application/json |
| DELETE | https://hostname/mga/sps/mga/user/mgmt/grant/{id} | There is no JSON payload included in the request. | <pre>{"result": message}</pre> <p>If the request completes successfully, the HTTP response code is 200.</p> <p>If the request does not complete successfully, the HTTP response is 500.</p> | application/json |
| PUT | https://hostname/mga/sps/mga/user/mgmt/grant/{id} | <pre>", "isEnabled": {true false}}</pre> | <pre>{"result": message}</pre> <p>If the request completes successfully, the HTTP response code is 200.</p> | application/json |

“Managing OAuth 2.0 authorization grants” on page 98
 You can view your authorization grants and the tokens and attributes of each authorization grant.

Chapter 11. Modifying template files

Use the local management interface to manage files and directories in the template files.

About this task

You can run the following tasks on the template files:

- **Edit**- Use this option if you want to view or modify the template file.
- **Import**- Use this option if you to import a file to the template files root.
- **Export**- Use this option if you want to export a file from the template files root.
- **Import Zip**- Use this option if you want to import the template files from a compressed file.
- **Export Zip**- Use this option if you want to export the template files as a compressed file.

Note: You cannot add newly created files to the set of predefined template files. You cannot import or export new files. A suggested best practice is to host static content on a web server, which can be loaded by the template files.

Default template files can be altered between updates. Deployed template files are not changed between upgrades to ensure modified customer template files are not affected. Updated template files are available for download by navigating to **Manage System Settings > Secure Settings > File Downloads** section of the appliance. Once downloaded, a customer can then use these files to modify and update existing template files.

Procedure

1. From the top menu, select **Secure Mobile Settings > Manage > Template Files**.
2. Work with all the management files and directories.

View or update the contents of a file in the template files root

- a. Select the file of interest.
- b. Select **Edit**. You can then view the contents of the file.
- c. Edit the contents of the file.
- d. Click **Save**.

Export a file from the template files root

- a. Select the file of interest.
- b. Select **Manage > Export**.
- c. Confirm the save operation when your browser displays a confirmation window.

Import a file to the template files root

- a. Select the file of interest.
- b. Select **Manage > Import**.
- c. Click **Browse**.
- d. Browse to the file you want to import.
- e. Click **Open**.
- f. Click **Import**.

Export the template file as a compressed file

- a. Select **Manage > Export Zip**.
- b. Confirm the save operation when your browser displays a confirmation window.

Import the template files as a compressed file

Make sure that the .zip file contains files that exist in the document root.

- a. Select **Manage > Import Zip**.
- b. Click **Browse**.
- c. Browse to the file you want to import.
- d. Click **Open**.
- e. Click **Import**.

Note: The file is not imported, and no action occurs if you export a template page file and immediately try to import a file. After you export a file, refresh the browser before you try to import a file.

3. When you edit or import template files, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.

For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Default template files

A list of the default template files and descriptions.

Table 8. Default template files in the ac/ directory

| File name | Description |
|---|---|
| ac/consent-form.html | Consent to register device form |
| ac/info.js | Detects the location of the device from which the requests are made. Collects the web browser attributes and sends them to the server for storing in the database. When the user logs out or when the current session times out, the script deletes the attributes from the database. |
| ac/javascript_rules/dynamic_attributes.js | This script runs after each request is processed by risk engine. It allows you to capture attributes that do not get captured automatically. Captured attributes are stored either in the session storage or the behavior storage area of the risk-based component, or both. The risk profile configuration dictates where the attributes are stored. |

Table 9. Default template files in the mga/ directory

| File name | Description |
|-------------------------|---|
| mga/user/mgmt/common.js | Script used by one-time password pages and by device management pages. Contains functions and properties used for making calls to the user self-care REST services. |

Table 9. Default template files in the mga/ directory (continued)

| File name | Description |
|---|--|
| mga/user/mgmt/device/device_attributes.html | Enables or disable devices. Renames or removes device. Displays all of the attributes for a device. |
| mga/user/mgmt/device/device_attributes.js | Script to process values entered in the device_attributes.html template |
| mga/user/mgmt/device/device_selection.html | Displays device name, status, (enabled or disabled) and time of last activity. |
| mga/user/mgmt/device/device_selection.js | Script to process data to display in the device_selections.html template |
| mga/user/mgmt/otp/otp.html | HMAC One Time Password Shared Key template page. This page enables the user too reset TOTP and HOTP Secret Key |
| mga/user/mgmt/otp/otp.js | Script to reset TOTP and HOTP Secret Key |

Table 10. One-time password template files

| File name | Description and link to file contents |
|--|--|
| otp/change_pin.html | This template file enables the user to enter a new pin. |
| otp/delivery/email_message.xml | This template page is used by EmailOTPDelivery as the content of the email that it sends to the user. |
| otp/delivery/sms_message.xml | This template page is used by SMSOTPDelivery as the content of the SMS that it sends to the user. |
| otp/delivery_selection.html | This template page displays the list of methods for generating, delivering, and verifying the one-time password. |
| otp/errors/allerror.html | This template page displays general errors that happen during the one-time password flow. |
| otp/errors/error_could_not_validate_otp.html | This template page displays errors during the validation of the one-time password that the user submits. |
| otp/errors/error_generating_otp.html | This template page displays errors during the generation of a one-time password. |
| otp/errors/error_get_delivery_options.html | This template page displays errors during the retrieval of the list of methods for delivering one-time password to the user. |
| otp/errors/error_otp_delivery.html | This template page displays errors during the delivery of a one-time password to the user. |
| otp/errors/error_sts_invoke_failed.html | This template page displays errors during the invocation of the Security Token Service. |
| otp/login.html | This template page displays the form where the user can enter the one-time password. |
| otp/next_otp.html | This template page enables the user to enter the next one time password. |

Table 11. Default files in the proper/ directory

| File name | Description |
|--|---|
| proper/errors/access_denied.html | This template page displays a message that the user cannot access the requested resource. |
| proper/errors/allerror.html | This template page displays general errors that are not displayed in other template files. |
| proper/errors/missingcomponent.html | This template page displays an error stating that the component required to process the request was not correctly configured or could not be initialized. |
| proper/errors/need_authentication.html | This template page displays an error stating that authentication is required in order to access the requested resource. |
| proper/errors/noprotdet.html | This template displays an error stating that the access request is to an unknown address, which might be because no configured endpoint or protocol exists that is mapped to this endpoint. |
| proper/errors/protocol_error.html | This template displays errors stating that an error occurred fulfilling a request to a specified address, and the error was caused by an unexpected error on the invoked protocol module. |

Table 12. OAuth template files

| File name | Description |
|------------------------------|---|
| oauth20/clients_manager.html | This template page is used by resource owners to show and manage trusted clients information. |
| oauth20/user_consent.html | This template page is used by the authorization server to determine and store user consent information about which OAuth clients are authorized to access the protected resource. This page also indicates scopes that the OAuth client requests. |
| oauth20/user_error.html | This template page shows detailed text information when an error occurs in an OAuth 2.0 flow. |
| oauth20/user_response.html | This template page displays the authorization code of an OAuth client that did not specify a client redirection URI upon partner registration. |

Default template files language support

A list of the languages that are supported by the default template files.

Table 13. Directory names and supported languages

| Directory name | Language |
|----------------|----------|
| ar | Arabic |
| cs | Czech |

Table 13. Directory names and supported languages (continued)

| Directory name | Language |
|----------------|---------------------|
| de | German |
| es | English |
| fr | French |
| hu | Hungarian |
| it | Italian |
| ja | Japanese |
| ko | Korean |
| pl | Polish |
| pt_BR | Portuguese (Brazil) |
| ru | Russian |
| zh_CN | Simplified Chinese |
| zh_TW | Traditional Chinese |

Chapter 12. Managing advanced configuration

Adjust configuration settings in supported configuration files.

About this task


The advanced configuration panel displays a table of configuration settings. Some can be modified and some are read-only. Each setting is displayed as a row in the table. The name of the setting is listed in the *key* column. The current value of the key is listed in the *value* column.

You can locate a setting by using one of the following methods:

- Scroll through the list until you see the setting.
By default, all configuration settings are included in the list.
- Filter the list by entering a string in the **Filter** field.
When you enter a string, the list is modified to show only the settings that contain the specified string.
- Filter the list by selecting a category from the **Filter by Category** menu. For descriptions of the categories and properties, see “Advanced configuration properties” on page 108.

Procedure

1. Select **Secure Mobile Settings > Manage > Advanced Configuration**.
2. To edit a key, select the edit icon  for the key.

Note: You cannot edit keys that are marked with the read-only icon: .

When you choose to edit a key, a new window displays the name of the key and the current value.

3. Edit the value as appropriate for your deployment.

Table 14. Configuration data types

| Data type | Action |
|-----------|---|
| Integers | Use the arrow icons to increment or decrement the value. Alternatively, you can type a new value in the text field. |
| Strings | Enter a string value in the text field. Note: Text fields must have a value. You cannot specify an empty field. To clear values from a field, enter NULL. |
| Booleans | Select the check box to set the value to true. Clear the check box to set the value to false. |

4. Click **OK**.
5. When you edit a property, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.
For more information, see Chapter 13, “Deploying pending changes,” on page 119.

Advanced configuration properties

Modify the advanced configurations to meet the requirements of your organization.

Category filter

The category filter displays names of grouping of configuration settings. The groupings correspond to functional areas. When you select a category, the user interface displays only the settings for the category.

Table 15. Filter by Category

| Category | Displays values for: |
|-------------------------|---|
| All | All keys |
| poc.signIn | "WebSEAL Sign-In Callback" on page 109 |
| poc.localIdentity | "WebSEAL Local Identity Callback" on page 109 |
| poc.websealAuth | "WebSEAL Authenticate Callback" on page 110 |
| poc.otpAuth | "One-time password Authenticate Callback" on page 110 |
| poc.authPolicy | "Authentication-Policy Callback" on page 110 |
| sps.httpRequestClaims | "SPS HTTP request claims" on page 111 |
| distributedMap | "Distributed shared data storage" on page 111 |
| userBehavior | "Attribute matcher properties" on page 111 |
| ipReputation | "PIP properties" on page 112 |
| attributeCollection | "Attribute collector properties" on page 112 |
| deviceRegistration | "Device registration properties" on page 113 |
| runtime | "Runtime properties" on page 114 |
| sps.page | "SPS page" on page 115 |
| riskEngine | "Risk engine properties" on page 115 |
| sps.authService | "Authentication service properties" on page 115 |
| distributedSessionCache | "Distributed session cache" on page 116 |
| otp.hotp.advanced | "HOTP advanced" on page 116 |
| otp.totp.advanced | "TOTP advanced" on page 116 |
| otp.retry | "TOTP and HOTP retry properties" on page 117 |
| otp.rsa.advanced | "RSA advanced" on page 117 |
| audit.advanced | "Audit advanced" on page 117 |
| oauth20 | "OAuth20" on page 117 |
| util.httpClient | "HTTP client" on page 118 |

WebSEAL Sign-In Callback

poc.signIn.attributesResponseHeader

The name of the header that contains the attributes of the user.

Data type: String

Example: am-fim-eai-xattrs

poc.signIn.credResponseHeader

The name of the header that contains the IVCred of the user.

Data type: String

Example: am-fim-eai-pac

poc.signIn.groupsResponseHeader

The name of the header that contains the groups of the user.

Data type: String

Example: fim.groups

poc.signIn.serverResponseHeader

The name of the header that contains the hostname that authenticates the user.

Data type: String

Example: fim.server

poc.signIn.targetResponseHeader

The name of the header that contains the redirect URL.

Data type: String

Example: am-fim-eai-redir-url

poc.signIn.userRequestHeader

The name of the header that contains the user name of the user.

Data type: String

Example: iv-user

poc.signIn.userResponseHeader

The name of the header that contains the user name of the user.

Data type: String

Example: am-fim-eai-user-id

poc.signIn.userSessionResponseHeader

The name of the header that contains the authentication level of the user.

Data type: String

Example: am-eai-auth-level

WebSEAL Local Identity Callback

poc.localIdentity.attributesRequestHeader

The name of the header that contains the attributes of the user.

Data type: String

Example: fim.attributes

poc.localIdentity.credRequestHeader

The header that contains the IVCred of the user.

Data type: String

Example: iv-creds

poc.localIdentity.groupsRequestHeader

The name of the header that contains the groups of the user.

Data type: String

Example: iv-groups

poc.localIdentity.userRequestHeader

The name of the header that contains the user name of the user.

Data type: String

Example: iv-user

WebSEAL Authenticate Callback

poc.websealAuth.authLevel

The authentication level of the callback.

Data type: Integer

Example: 1

poc.websealAuth.userRequestHeader

The name of the header that contains the user name of the user.

Data type: String

Example: iv-user

One-time password Authenticate Callback

poc.otp.authLevel

The authentication level of the callback.

Data type: Integer

Example: 2

Authentication-Policy Callback

poc.authPolicy.allowRequestOverride

Whether the authentication level, the authentication mode, and the authentication type of the callback can be overwritten by query string parameters.

Data type: Boolean

Example: true

poc.authPolicy.authLevel

The authentication level of the callback.

Data type: Integer

Example: 1

poc.authPolicy.authType

The authentication type of the callback.

Data type: String

Example: COMPLEMENTARY, HIERARCHICAL

SPS HTTP request claims

sps.httpRequestClaims.enabled

Whether HTTP request information is sent to STS as HTTPRequestClaims.

Data type: Boolean

Example: false

sps.httpRequestClaims.filterSpec

The filter that specifies the HTTP request information that is sent to STS as HTTPRequestClaims.

Data type: String

Example: cookies=*:headers=*

Distributed shared data storage

distributedMap.cleanupWait

The amount of time, in milliseconds, to wait before it performs another cleanup against the distributed map.

Data type: Integer

Example: 10000

distributedMap.defaultTTL

The amount of time, in seconds, that the entries in the distributed map must live when no lifetime is specified for an entry.

Data type: Integer

Example: 3600

distributedMap.getRetryDelay

The amount of time, in milliseconds, to wait before it performs another retrieval against the distributed map.

Data type: Integer

Example: 500

distributedMap.getRetryLimit

The number of retrievals that is done against the distributed map before it returns that the retrieved data is not in the distributed map.

Data type: Integer

Example: 10

Attribute matcher properties

userBehavior.minimumUsageHistoryRequired

Minimum usage data records required for any usage data analysis; used by LoginTimeMatcher.

Data type: Integer

Example:

8

userBehavior.ipAddressRequestAttribute

The XACML request attribute to read from the IP address.

Data type: String

Example:

urn:ibm:security:subject:ipAddress

PIP properties

ip.reputation.ipAddressAdverseReputationThreshold

The value that an IP classification score must be at or above for an IP address to be considered as that classification.

Data type: Integer

Example:

50

Attribute collector properties

attributeCollection.cookieName

Correlation ID used by the attribute collector.

Data type: String

Example:

ac.uuid

attributeCollection.requestServer

Request server for attribute collector. A list of the allowable hosts where the ajaxRequest can be sent from.

Data type: String List

Example:

https://rbademo.example.com,https://rbaemo2.example.com

attributeCollection.serviceLocation

Location of the attribute collector.

Data type: String List

Example:

http://rbademo.example.com/mga

attributeCollection.sessionTimeout

Number of seconds in which sessions stored in context-based access will automatically expire, unless updated. If any attribute in the session is updated, the session expiry is extended by the specified number of seconds configured in this property. The default is 3600 seconds.

Data type: Integer

Example:

1800 seconds

attributeCollection.enableGetAttributes

Enables the REST GET method to return attributes.

Data type: Boolean

Example:

False

attributeCollection.getAttributesAllowedClients

A comma-separated list of clients allowed to access the ACS REST GET method.

If this property is not set and `attributeCollection.enableGetAttributes` is set to true, anyone can access the GET method. If this property is set but `attributeCollection.enableGetAttributes` is set to false, this property is ignored.

Data type: String List

Example:

hostname1, hostname2

attributeCollection.hashAlgorithm

The algorithm used to create the hash.

Data type: String

Example:

SHA256

attributeCollection.attributesHashEnabled

A comma-separated list of attribute URI values that have been configured for hashing.

Attention: Do not hash the following attributes:

- ipAddress
- geoLocation
- accessTime

Data type: String List

Example:

urn:ibm:security:environment:http:userAgent,
urn:ibm:security:environment:deviceFonts,
urn:ibm:security:environment:browserPlugins

attributeCollection.authenticationContextAttributes

Comma-separated lists of attribute names to be collected when performing an authentication service obligation.

Data type: String List

Example:

authenticationLevel, http:host

Device registration properties

deviceRegistration.maxRegisteredDevice

Maximum device fingerprint count. The default is 10. Valid values are 1 to 20.

Data type: Integer

Example:

10

deviceRegistration.maxUsageDataPerUser

Maximum number of historical usage attribute records stored per user. The default is 200. Valid values are 1 to 5000.

Data type: Integer

Example:

1000

deviceRegistration.deviceMatchThreshold

The risk score threshold where an existing fingerprint is considered to match the incoming device fingerprint.

Data type: Integer

Example:

20

deviceRegistration.allowIncompleteFingerprints

Specifies to allow the device registration obligation to store fingerprints where all the fingerprint attributes are not available on the session information.

Data type: Boolean

Example:

False

deviceRegistration.permitOnIncompleteFingerprints

Specifies to permit access to the resource if the fingerprint collected by the device registration obligation does not include all fingerprint attributes.

Data type: Boolean

Example:

False

deviceRegistration.authLevelHeaderEnabled

Enables the consent to device registration authentication obligation to set the authentication level on the session.

Data type: Boolean

Example:

True

deviceRegistration.authLevelHeaderValue

The authentication level value to be used when the consent to device registration is configured to set the authentication level.

Data type: Integer

Example:

2

Runtime properties**runtime.dbLoggingEnabled**

Enables fine-grained logging for database SQL statements.

Data type: Boolean

Example:

False

runtime.hashAlgorithm

The algorithm that is used for hashing. The supported algorithms are:

- SHA-1
- SHA-256
- SHA-512

The default is SHA-256.

Data type: String

Example:

SHA-256

SPS page

sps.page.htmlEscapedMacros

A comma-separated list of macros that is HTML-escaped when it is rendered in pages that are sent to the browser.

Data type: String

Example:

```
@REQ_ADDR@,  
@DETAIL@,  
@EXCEPTION_STACK@,  
@EXCEPTION_MSG@,  
@OTP_METHOD_ID@,  
@OTP_METHOD_LABEL@,  
@OTP_HINT@,  
@ERROR_MESSAGE@,  
@MAPPING_RULE_DATA@
```

sps.page.exceptionMacros

A comma-separated list of classname:macro pairs. Classname is the fully qualified name of the exception class. Macro is the name of the macro to which the class maps.

Data type: String

Example:

```
com.tivoli.am.fim.otp.deliveries.OTPDeliveryException =  
  @OTP_DELIVERY_EXCEPTION@,  
com.tivoli.am.fim.otp.providers.OTPProviderException =  
  @OTP_PROVIDER_EXCEPTION@
```

Risk engine properties

riskEngine.reportsEnabled

Enables the generation of risk calculation reports.

Data type: Boolean

Example:

False

riskEngine.reportsMaxStored

Specifies the maximum number of reports to store.

Data type: Integer

Example:

5

Authentication service properties

sps.authService.reauthenticationEnabled

Specifies that the authentication service performs authentication even if the user already has an authenticated session at the required authentication level.

Data type: Boolean

Example:

true

Distributed session cache

distributedSessionCache.enabled

A switch that dictates if the distributed session cache is used for session failover. If this setting is not enabled, the distributed session cache server still runs as a service, but the client does not use it.

Data type: Boolean

Example: false

distributedSessionCache.localCacheSize

The number of sessions to be stored on the client as a local cache. A value of 0 or less means that any number of sessions can be cached by the client. A low number requires more connections to the distributed session cache if there are many active sessions. A high number runs the risk of running out of memory if many sessions are locally cached. All sessions are still stored on the distributed session cache when it is enabled.

Data type: Integer

Example: 4096

HOTP advanced

hotp.secretKeyAttributeName

The attribute name that is used for storage of the HOTP secret key in the database.

Data type: String

Example: otp.hmac.hotp.secret.key

hotp.secretKeyAttributeNamespace

The attribute namespace of the HOTP secret key. The namespace in combination with the attribute name constitutes the unique identifier for the attribute in the database.

Data type: String

Example: urn:ibm:security:otp:hmac

TOTP advanced

totp.secretKeyAttributeName

The attribute name that is used for storage of the TOTP secret key in the database.

Data type: String

Example: otp.hmac.totp.secret.key

totp.secretKeyAttributeNamespace

The attribute namespace of the TOTP secret key. The namespace and the attribute name constitute the unique identifier for the attribute in the database.

Data type: String

Example: urn:ibm:security:otp:hmac

TOTP and HOTP retry properties

otp.retry.enabled

Whether or not the retry protection is enabled.

Data type: Boolean

Example: true

otp.retry.maxNumberOfAttempts

The maximum number of strikes the user can have before being prevented from logging in.

Data type: Integer

Example: 5

otp.retry.otpRetryTimeout

The number in seconds a strike lasts.

Data type: Integer

Example: 600

RSA advanced

rsa.sessionTimeout

The length of time, in seconds, that a connection to the RSA Authentication Manager server remains open before it times out when a user attempts to authenticate.

Data type: Integer

Example: 1800

Audit advanced

audit.verboseEvents.enabled

Enables the inclusion of additional information in an audit event when the `audit.verboseEvents.enabled` property is set to true.

Note: The `audit.verboseEvents.enabled` property defaults to false.

Data type: Boolean

Example: true

OAuth20

oauth20.sessionEndpointEnabled

Enables the ability to return an authenticated session at the point-of-contact when the `oauth20.sessionEndpointEnabled` property is set to true.

Note: The `oauth20.sessionEndpointEnabled` property defaults to false.

Data type: Boolean

Example: false

oauth20.tokenCache.cleanupWait

The amount of time, in seconds, to wait before it performs another cleanup of expired tokens in the OAuth 2.0 token cache.

Note: The `oauth20.tokenCache.cleanupWait` property defaults to 120.

Data type: Integer

Example: 120

oauth20.doNotSendXFrameOptionsHeader

Specifies whether an X-Frame-Options header with value SAMEORIGIN must be returned from the OAuth 2.0 endpoints. When set to true, no X-Frame-Options header is sent.

Note: The `oauth20.doNotSendXFrameOptionsHeader` property defaults to false.

Data type: Boolean

Example: false

HTTP client

util.httpClient.defaultTrustStore

Stores the default truststore that HTTPS connections in HTTP client uses.

Note: The `util.httpClient.defaultTrustStore` property defaults to `rt_profile_keys`.

Data type: String

Example: `rt_profile_keys`

Chapter 13. Deploying pending changes

Some configuration and administration changes require an extra deployment step.

About this task

When you use the graphical user interface on the appliance to specify changes, some configuration and administration tasks take effect immediately. Other tasks require a deployment step to take effect. For these tasks, the appliance gives you a choice of deploying immediately or deploying later. When you must make multiple changes, you can wait until all changes are complete, and then deploy all of them at one time.

When a deployment step is required, the user interface presents a message that says that there is an undeployed change. The number of pending changes is displayed in the message, and increments for each change you make.

Note: If any of the changes require the runtime server to be restarted, the restart occurs automatically when you select **Deploy**. The runtime server will then be unavailable for a period of time until the restart completes.

Procedure

1. When you finish making configuration changes, select **Click here to review the changes or apply them to the system**.

The Deploy Pending Changes window is displayed.

2. Select one of the following options:

| Option | Description |
|------------------|---|
| Cancel | Do not deploy the changes now. Retain the undeployed configuration changes. The appliance user interface returns to the previous panel. |
| Roll Back | Abandon configuration changes. A message is displayed, stating that the pending changes were reverted. The appliance user interface returns to the previous panel. |
| Deploy | Deploy all configuration changes. When you select Deploy , a system message is displayed, stating that the changes were deployed. If any of the changes require the runtime server to be restarted, the restart occurs automatically when you select Deploy . The runtime server will then be unavailable for a period of time until the restart completes. |

Chapter 14. Tuning runtime application parameters and tracing specifications

To manually tune selected runtime application parameters and tracing specifications, use the Runtime Parameters management page.

Procedure

1. From the top menu, select **Secure Mobile Settings > Manage > Runtime Parameters**. This page contains three panels: **Runtime Status**, **Runtime Tuning Parameters**, and **Runtime Tracing**.
2. Perform one or more of the following actions to tune your runtime.

Note: Certain changes might require a restart of the runtime before they can take effect.

View the status of and restart the runtime

- a. Select the **Runtime Status** panel. The status of local and clustered runtimes are displayed.
 - Under **Local Runtime Status**, you can view the runtime operational status, when it was last started, and whether a restart is outstanding. If the value of the **Restart Required** field is **True**, it means that the runtime must be restarted for some changes to take effect.
 - Under **Clustered Runtime Status**, all nodes in the cluster are listed.
 - The **Master** column indicates whether a node is the cluster master.
 - The **Runtime Status** column indicates whether a node is running or stopped.
 - The **Changes Active** column indicates whether changes made to the cluster configuration are active on this node. Having a green indicator in this column means that all changes made are already active. Having a yellow indicator in this column means that this node must be restarted before some changes can take effect.
- b. Depending on which runtime you want to restart, click **Restart Local Runtime** or **Restart All Clustered Runtimes**.

Modify the maximum or initial heap size

These parameters indicate the maximum and initial heap size in megabytes for the runtime Java virtual machine.

- a. On the **Runtime Tuning Parameters** panel, select **Max Heap Size** or **Initial Heap Size**.
- b. Click **Edit**.
- c. In the Max Heap Size or Initial Heap Size window, enter the heap size value as needed.
- d. Click **OK**.

Modify whether to suppress sensitive trace

Enabling this parameter prevents sensitive information from being exposed in log and trace files. Examples of such sensitive information include bytes received over a network connection.

- a. On the **Runtime Tuning Parameters** panel, select **Suppress Sensitive Trace**.
- b. Click **Edit**.
- c. In the Suppress Sensitive Trace window, select or clear the check box as needed.
- d. Click **OK**.

Modify console log level

Console log level controls the granularity of messages that go to the console.log file.

- a. On the **Runtime Tuning Parameters** panel, select **Console Log Level**.
- b. Click **Edit**.
- c. In the Console Log Level window, select the new value from the list.
- d. Click **OK**.

Set whether to accept client certificates

This parameter controls whether the server accepts client certificates as a form of authentication.

- a. On the **Runtime Tuning Parameters** panel, select **Accept Client Certificates**.
- b. Click **Edit**.
- c. In the Accept Client Certificates window, select or clear the check box as needed.
- d. Click **OK**.

Set session invalidation timeout

This parameter defines the amount of time a session can remain unused before it is no longer valid.

- a. On the **Runtime Tuning Parameters** panel, select **Session Invalidation Timeout**.
- b. Click **Edit**.
- c. In the Session Invalidation Timeout window, define the value in seconds.
- d. Click **OK**.

Set session reaper poll interval

This parameter defines the wake-up interval in seconds for the process that removes invalid sessions. The minimum value is 30 seconds. If a value less than the minimum is entered, an appropriate value is automatically determined and used. This value overrides the default installation value, which is 30 - 360 seconds, based on the session timeout value. Because the default session timeout is 120 minutes, the reaper interval is usually 2 - 3 minutes.

- a. On the **Runtime Tuning Parameters** panel, select **Session Reaper Poll Interval**.
- b. Click **Edit**.

- c. In the Session Reaper Poll Interval window, define the value in seconds.
- d. Click **OK**.

Set the keystore that is used by the runtime server

This parameter defines the key database that contains the runtime server's private key.

- a. On the **Runtime Tuning Parameters** panel, select **Keystore**.
- b. Click **Edit**.
- c. In the Keystore window, select the key database from the list.
- d. Click **OK**.

Set the truststore that is used by the runtime server

This parameter defines the key database that contains keys that are trusted by the runtime server

- a. On the **Runtime Tuning Parameters** panel, select **Truststore**.
- b. Click **Edit**.
- c. In the Truststore window, select the key database from the list.
- d. Click **OK**.

Delete the value of a parameter

Use this button to delete the existing value of a parameter.

- a. Select the parameter to reset the value for.
- b. Click **Delete**. The value of the parameter is then changed to Unset.

Manage the application interface on which the runtime listens

- a. On the **Runtime Tuning Parameters** panel, under **Runtime Listening Interfaces**, you can add, edit, or delete a listening interface.

To add a listening interface

- 1) Click **Add**.
- 2) In the Runtime Listening Interfaces window, select the listening interface from the list.
- 3) Specify the listening port.
- 4) Select the **SSL** check box if security is required.
- 5) Click **OK**.

To modify a listening interface

- 1) Select the listening interface to edit.
- 2) Click **Edit**.
- 3) In the Runtime Listening Interfaces window, edit the values as needed.
- 4) Click **OK** to save the changes.

To delete a listening interface

- 1) Select the listening interface to delete.
- 2) Select **Delete**.
- 3) Confirm the deletion.

Manage tracing specification

- a. Select the **Runtime Tracing** link from the top of this page. You can also access this panel from the top menu by selecting **Monitor Analysis and Diagnostics > Logs > Runtime Tracing**.
- b. Use one of the following ways to edit the trace level of a component.
 - Select the component name from the **Component** list. Select the ideal trace level for this component from the **Trace Level** list. Then, click **Add**. Repeat this process to modify trace levels for other components if needed. To clear all of the tracing levels, click **Clear**.

To log all events, select ALL as the trace level.

Note: This setting increases the amount of data in logs, so use this level when necessary.

```
com.ibm.tsc.rtss.*=ALL:com.ibm.sec.authz.*=ALL  
com.tivoli.am.fim.trustserver.sts.modules.*=ALL:  
com.tivoli.am.fim.otp.*=ALL  
com.tivoli.am.rba.*=ALL
```

- Enter the name and value of the trace component in the **Trace Specification** field. To modify multiple components, separate two strings with a colon (:). Here is an example.

```
com.x.y.*=WARNING:com.a.b.*=WARNING:com.ibm.isam.*=INFO
```

- c. Click **Save**.
3. When you make changes, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.

Chapter 15. Options for handling session failover events

Security Access Manager for Mobile offers several solutions to the challenge of providing sharing of session state across multiple servers in a clustered environment.

The following sections describe the options available for handling failover events in clustered environments:

- No handling of failover events
- The Distributed Session Cache

Option 1: No handling of failover events

Failover events are rare when WebSEAL or Web Reverse Proxy instance is configured to maintain session affinity in a stateful junction to Security Access Manager for Mobile.

This scenario is applicable in the case of using the `isamcfg` tool to configure the junction.

When failover events do occur, session state is lost and clients might be required to restart their current transaction.

This option is configured by default. However, there is a risk of a poor user experience when:

- A Security Access Manager for Mobile server becomes unavailable
- The WebSEAL or Web Reverse Proxy cannot maintain session affinity

Option 2: The distributed session cache

The distributed session cache (DSC) can be used for session storage by all Security Access Manager appliances in a cluster.

When a fail over event occurs, Security Access Manager appliance retrieves the session data of the user from the DSC. It therefore maintains the existing session state.

Within Security Access Manager, the DSC is part of the cluster configuration. For more information about turning on or turning off this feature, see the Distributed Session Cache section in "Advanced configuration properties" on page 108.

For more information about the distributed session cache and cluster configuration, search for the "Managing cluster configuration" topic in the *IBM Security Access Manager Appliance Administration Guide*.

Chapter 16. Call Java code from within JavaScript rules

IBM Security Access Manager for Mobile JavaScript rules supports calling Java code from within JavaScript. The set of classes that can be called are restricted.

Exercise reasonable caution when calling Java code from JavaScript rules to ensure that accidental damage to appliance resources is avoided.

Access of the JavaScript mapping rule to the Java classes are limited to the following classes:

- `com.tivoli.am.fim.trustserver.sts.STSModuleException`
- `com.tivoli.am.fim.trustserver.sts.STSUniversalUser *`
- `com.tivoli.am.fim.trustserver.sts.uuser.Attribute *`
- `com.tivoli.am.fim.trustserver.sts.uuser.AttributeList *`
- `com.tivoli.am.fim.trustserver.sts.uuser.AttributeStatement *`
- `com.tivoli.am.fim.trustserver.sts.uuser.ContextAttributes *`
- `com.tivoli.am.fim.trustserver.sts.uuser.Group *`
- `com.tivoli.am.fim.trustserver.sts.uuser.Principal *`
- `com.tivoli.am.fim.trustserver.sts.uuser.Subject *`
- `com.tivoli.am.fim.trustserver.sts.uuser.RequestSecurityToken *`
- `com.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils`
- `com.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtCacheDMPImpl`
- `com.tivoli.am.fim.trustserver.sts.utilities.InfoCardClaim`
- `com.tivoli.am.fim.trustserver.sts.utilities.QueryServiceAttribute`
- `com.tivoli.am.fim.trustserver.sts.utilities.USCContextAttributesHelper`
- `com.tivoli.am.fim.base64.BASE64Utility`
- `com.tivoli.am.fim.utils.IteratorWrapper`
- `com.tivoli.am.rba.attributes.AttributeIdentifier***`
- `com.tivoli.am.rba.extensions.RBAExtensions***`
- `com.tivoli.am.rba.rtss.AttributeLocatorImpl***`
- `com.tivoli.am.rba.fingerprinting.ValueContainerIdentifierAdapter***`
- `com.ibm.ws.logging.internal.impl.BaseTraceService$TeePrintStream***`
- `java.lang.Character +`
- `java.lang.Object +`
- `java.lang.String +`
- `java.lang.reflect.Array +`
- `java.io.ByteArrayInputStream +`
- `java.io.ObjectInputStream +`
- `javax.mail.internet.InternetAddress +`
- `java.util.ArrayList **+`
- `java.util.HashSet **+`
- `java.util.HashMap **+`
- `java.io.PrintStream***`
- `java.lang.System***`

Notes:

* The white list does not contain any implementation of the interfaces that are defined in the `org.w3c.dom` package. For example, you cannot use the method `org.w3c.dom.Document toXML()` in `com.tivoli.am.fim.trustserver.sts.STSUniversalUser`.

** Inner classes are not supported. Methods that involve an inner class implementation of an interface are not available. For example, do not use the following method in `java.util.HashMap`:

- `Collection<V> values()`
- `Set<K> keySet()`
- `Set<Map.Entry<K,V>> entrySet()`

*** These classes are only applicable to Dynamic attributes.

+ These classes are applicable to both Dynamic attributes and the customizable one-time password mapping rules.

You can customize the following one-time password mapping rules:

- “OTPDeliver mapping rule” on page 51
- “OTPGenerate mapping rule” on page 51
- “OTPGetMethods mapping rule” on page 50
- “OTPVerify mapping rule” on page 52

You can customize and update Dynamic attributes.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features contained in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM might have patents or pending patent applications that cover subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it to enable: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information might be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments might vary significantly. Some measurements might have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements might have been estimated through extrapolation. Actual results might vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding the future direction or intent of IBM are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs that conform to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample

programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2004, 2012. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations might not appear.

Privacy Policy Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: <http://www.ibm.com/legal/copytrade.shtml>

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

The Oracle Outside In Technology included herein is subject to a restricted use license and can only be used in conjunction with this application.

Index

A

- access token OAuth 67
- accessibility x
- activation 3
- advanced configuration
 - category filter 108
 - property descriptions 108
- API definition
 - API definition
 - attaching to resource 83
 - publishing 83
 - API protection
 - client 81, 82
 - definition 79
 - API protection client
 - managing 82
 - registering 81
 - API protection definition
 - creating 76
 - managing 79
- application interface
 - manage 5
- authentication
 - client 74
- authorization code OAuth 67
- authorization grant OAuth 67

C

- certificates
 - client
 - See client certificates
- client
 - managing API protection 82
 - registering API protection 81
- client authentication
 - OAuth 2.0 token endpoint 74
 - types 74
- client certificate authentication 13
- clients
 - OAuth 67
- cluster
 - cluster configuration management
 - page
 - LMI 17
 - configuration 17
 - master nodes
 - configure 17
 - registration 17
 - unregistration 17
 - cluster signature file
 - export 17
 - import 17
- compliance
 - NIST SP800-131a 31
- configuration
 - advanced 107

D

- definition
 - creating API protection 76
 - managing API protection 79
- deploying changes 119
- device fingerprints
 - managing 98
- distributed session cache 125

E

- education x
- endpoints
 - OAuth
 - definitions 68
 - URLs 68
- error messages
 - OAuth HTTP 97

F

- federation
 - OAuth 2.0
 - endpoint definitions 68
 - naming 68
 - URLs 68
 - OAuth configuration 76

G

- getting started
 - configure 1

I

- IBM
 - Software Support x
 - Support Assistant x
- isamcfg
 - command line reference 19
 - overview 15
 - reference 19
 - WebSEAL point of contact 16
- isamcfg tool
 - appliance 15
 - external machine 16
 - reverse proxy instance 16
 - Web Gateway Appliance 16
 - Web Gateway Appliance
 - configuration 21
 - WebSEAL configuration 25
- isamcfg worksheet
 - Web Gateway Appliance 21
 - WebSEAL 25
- isamcfg
 - reverse proxy instance 16
 - WebSEAL policy enforcement
 - point 16

L

- license, support 3
- LMI
 - cluster configuration management
 - page 17

M

- mapping rules
 - custom
 - one-time password 127
 - customizing for context data 54
 - managing 49
 - OTPDeliver 51
 - OTPGenerate 51
 - OTPGetMethods 50
 - OTPVerify 52
 - PostTokenGeneration 87

N

- NIST SP800-131a compliance 31
- notices 129

O

- OAuth
 - API protection client 81
 - endpoints 68
 - federation configuration 76
 - HTTP error responses 97
- OAuth 2.0
 - about 69
 - authorization code 67
 - concept 69
 - endpoint
 - definitions 68
 - URLs 68
 - overview 69
 - state management 74
 - template page types 91, 94, 95
 - token endpoint client
 - authentication 74
 - trusted clients management 75
 - workflow 69
- OAuth support 67
- one-time password
 - configuration 36, 37
 - configuring an RSA provider 43
 - configuring delivery 46
 - configuring HOTP 38
 - configuring MAC 41
 - configuring TOTP 40
 - customizing delivery 55
 - delivery method 50, 87
 - delivery methods 37
 - managing mapping rules 49
 - overview 35
 - providers 37

- online
 - publications ix
 - terminology ix
- OTPDeliver
 - usage 51
- OTPGenerate
 - usage 51
- OTPGetMethods
 - usage 50
- OTPVerify
 - usage 52

P

- pending changes 119
- point of contact
 - token endpoint 74
- policy
 - policy
 - attaching to resource 83
 - publishing 83
- PostTokenGeneration
 - usage 87
- problem-determination x
- protected resource OAuth 67
- publications
 - accessing online ix
 - list of for this product ix

R

- resource owner OAuth 67
- resource server OAuth 67
- rest services for authorization grant management 99
- rest services for otp secret keys 65
 - hotp secret keys 65
 - totp secret keys 65
- runtime component
 - configure 7
 - manage 7
- runtime security services EAS
 - update library file 11

S

- servers
 - OAuth 67
- specifications OAuth 67
- starting configuration
 - steps 1
- state management OAuth 2.0 74
- support license 3

T

- template files
 - default files 102
 - language support 104
- template files root
 - manage 55, 101
- template pages
 - consent to authorize 91
 - error 94

- template pages (*continued*)
 - examples
 - errors OAuth 2.0 94
 - response OAuth 2.0 94
 - trusted clients management 95
 - one-time password
 - delivery error 61
 - delivery selection 58
 - email 64
 - general errors 59
 - generating one-time password error 59
 - get delivery error 60
 - login 57
 - SMS 64
 - STS error 62
 - validation error 63
 - response 94
 - trusted clients management 95
- terminology ix
- tool
 - isamcfg 15
- training x
- troubleshooting x
- trusted clients management
 - overview 75
 - template pages 95

U

- user self-administration 98

W

- WebSEAL
 - update runtime security services EAS
 - library file 11



Product Number: 5725-L52

Printed in USA

SC27-6205-01

