IBM® Security Access Manager
for Versions 9.0.3

Memorial Hermann Authentication Plugin

Contents

Preface

Access to publications and terminology

The following publications complement the information contained in this document:

Publication Library

These publications complement the information that is contained in this publication:

Base Information

- *IBM® Tivoli® Access Manager Base Installation Guide*

  Explains how to install, configure, and upgrade Access Manager software, including the Web portal manager interface.

- *IBM Security Access Manager Base Administrator's Guide*

  Describes the concepts and procedures for using Access Manager services. Provides instructions for managing tasks from the Web portal manager interface and by using the pdadmin command.

WebSEAL Information

- *IBM Security Access Manager WebSEAL Installation Guide*

  Provides installation, configuration, and removal instructions for the WebSEAL server and the WebSEAL application development kit.

- *IBM Security Access Manager WebSEAL Administrator's Guide*

  Provides background material, administrative procedures, and technical reference information for using WebSEAL to manage the resources of your secure Web domain.

- *IBM Security Access Manager WebSEAL Developer's Reference*

  Provides administration and programming information for the Cross-domain Authentication Service (CDAS), the Cross-domain Mapping Framework (CDMF), and the Password Strength Module.

Web Gateway Appliance Information

- *IBM Security Access Manager Web Gateway Appliance Administration Guide*

  Provides information about configuring and maintaining a Security Access

Manager environment.

- *IBM Security Web Gateway Appliance Configuration Guide for Web Reverse Proxy*

  Provides configuration procedures and technical reference information for the Web Gateway Appliance.

- *IBM Security Web Gateway Appliance Web Reverse Proxy Stanza Reference*

  Provides a complete stanza reference for the Web Gateway Appliance Web Reverse Proxy.

Mobile Information

- *IBM Security Access Manager for Mobile Administration Guide*

  Describes how to manage, configure, and deploy an existing IBM Security Access Manager environment.

- *IBM Security Access Manager for Mobile Configuration Guide*

  Explains how to complete the initial configuration of IBM Security Access Manager for Mobile.

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. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

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`.

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.

Product name updates

This publication was first established for IBM Tivoli Access Manager. IBM Tivoli Access Manager has since been superseded by IBM Security Access Manager.

Wherever in this guide, any figures and graphics that contain or refer to IBM Tivoli Access Manager, the use of IBM Security Access Manager is implied. There are no functionality discrepancies between IBM Tivoli Access Manager and IBM Security Access Manager.

Introducing the Library

Introduction

The Memorial Hermann Authentication Plugin uses OAuth 2.0 feature of IBM Security Access Manager (ISAM) to extend authentication to ISAM from traditional Web channels to native desktop operating systems using a modified OAuth mapping rule and client library.

The OAuth mapping rule contains logic to bind one or more OAuth grants to a user, and includes registration, deregistration, locking, and time-based reauthentication. The static OAuth grants represent the fixed value encoded in a physical smart-card authentication credential (Static Token). Each time an OAuth Access Token (AT) or Refresh Token (RT) expires is it reissued with the same value by the mapping rules to remain consistent with the physical card value.

The custom client required to handle the authentication phase and to implement the logic of the OAuth 2.0 to Static Token is implemented in provided native platform library. This library is then able to be called from a native application or system event such as a command line application or Windows Credential Provider or Linux Pluggable Authentication Module.

The library requires input configuration of the ISAM host, OAuth details (endpoint, client_id) and the static token value to be provided as an AT or RT depending on the authentication phase.

No provision is made for the library to acquire the static token value from scanning an RFID card, a magnetic-stripe card, etc. and no cryptography is performed on the token itself.

Integration Product Contents

The integration solution is packaged as a compressed file. The package contains the following files:

File Name	Description
`isam_mh_auth_plugin.pdf`	This integration guide.
`esso/`	TAM ESSO Restful API (RAPI) for user account retrieval and sample invocation code
`headers/`	C based header file for compilation of applications using the authentication library
`isam/`	ISAM OAuth 2.0 Mapping Rule files
`lib/libisamtokenauth.so`	Linux C based authentication library

File Name	Description
`lib/libisamtokenauth.dll`	Windows Dynamic Linked Library C based authentication library
`lib/libisamtokenauth.lib`	Windows Static Library for C based authentication library
`sample/isamAuth`	Linux sample calling application
`sample/isamAuth.exe`	Windows sample calling application
`src/`	Exported Git repository containing source of the authentication library including API documentation (README.md)

Table 1: Integration Package contents

Before you start

This document details the flows implemented by the plugin and calling the at a high level in your environment.

This guide does not cover the configuration of the entire environment. In particular, the following product installations and configurations must already be complete:

- IBM Security Access Manager
  - IBM Security Access Manager Web Reverse Proxy
  - IBM Security Access Manager Advanced Access Control
- Windows Operating System
  - libcurl.dll
    This should be built from source available from the official site https://curl.haxx.se/.
    Pre-compiled versions are available; however, no warranty or guarantee is made.
    An unofficial/unsupported build can be obtained from http://www.confusedbycode.com/curl/#downloads.
    During validation, the curl-7.46.0-win64.zip was used.
  - VCRUNTIME140.dll
    The Microsoft Visual C++ Runtime is available as a redistributable download from https://www.microsoft.com/en-us/download/details.aspx?id=53840
- Linux Operating System

  - libcurl.so
    Many Linux systems will have this library if cURL is installed.
    You can use the package manager for the distribution to check and install as necessary.

    For example, on RedHat Enterprise Linux

    ```
    [user@host ~]$ rpm -qa | grep curl
    curl-7.29.0-25.el7.centos.x86_64
    libcurl-7.29.0-25.el7.centos.x86_64
    ```

  - glibc-2.14 or above
    Install the GLIBC binary or package distribution.
    You can use the package manager for the distribution to check and install as necessary.

    For example, on RedHat Enterprise Linux

    ```
    [user@host ~]$ rpm -qa | grep glibc
    glibc-2.17-105.el7.x86_64
    glibc-common-2.17-105.el7.x86_64
    ```

Throughout this document, the following hostnames are used to represent the different interfaces of the ISAM appliance

- www.myidp.ibm.com - Web Reverse Proxy
- isam.myidp.ibm.com - Local Management Interface (LMI)

Solution Design

This authentication plug-in ties one or more "static tokens" to a user, and includes registration, deregistration, locking, and time-based reauthentication.

A custom client must be written to handle the client side authentication challenge. This design will include a high-level description of what it should do, but no implementation details. A sample application and source is provided as a reference sample as well as applicable cURL commands to simulate the flow in a controlling application or system call.

There must be some way for the user to scan the token into the client. The token could be an RFID card, a magnetic-stripe card, etc. No cryptography is performed on the token itself.

Design Overview

Token registrations are managed as long-lived OAuth 2.0 rants in the ISAM appliance that are bound to users. Creating a grant registers a token to the user; deleting a grant de-registers the token.

Enforcing a grant lifetime for Access Tokens enables the ability for time based reauthentication which only requires a physical credential to be tapped.

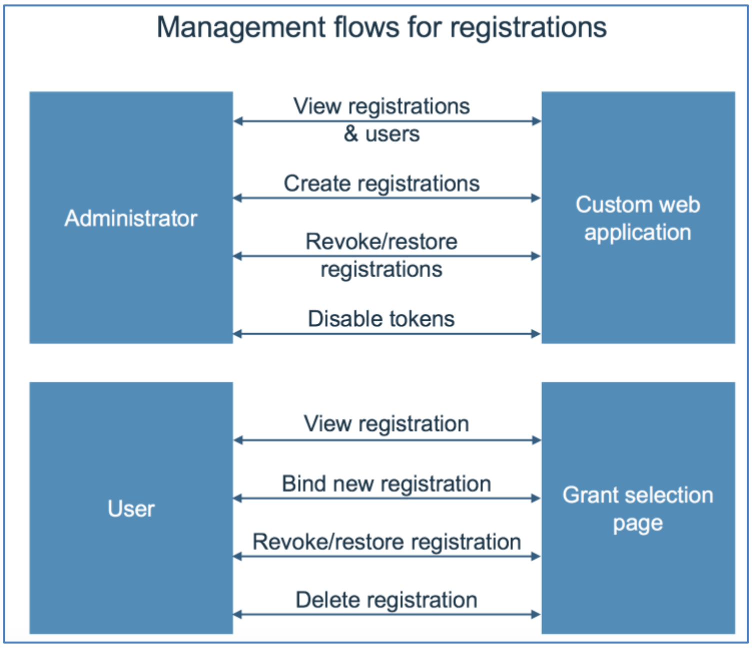
Authentication Flow	Interaction Requirement
Registration	Credential token tap/swipe Username and Password
Time Window (within N hours)	Credential token tap/swipe
Reauthentication (outside of N hours)	Credential token tap/swipe Password of the registered user

The provided library implements the flows detailed below. The client making use of the authentication library must implement the mechanism to read the data stored on the token.

The tokens are "static tokens", which encode data without performing cryptography. The data is processed on the client to generate unique identifiers for access token and refresh token. This process is deterministic and repeatable by every client, but the server doesn't need to know how it's calculated.

User self-care is provided through the existing ISAM USC features as well as management RAPIs.

Sample RAPI calls are provided for the Administrator functions and should be implemented in to an existing organisation support or security help desk portal or application.



Design Assumptions

1. Users have a maximum of one token registered at any one time
2. Tokens may be registered to a maximum of one user
3. Multiple concurrent login sessions are acceptable (i.e. valid access token based on the static token value)
4. Indefinite revocation of individual static tokens is required for lost or stolen cards.
5. Administration functions will be built in to an existing support/security/help desk system. Example RAPI calls are provided.
6. The authentication library only handles the authentication of a static token to activate that token for use for authentication. Example code with callback capability is provided.
7. No acquisition of the token value or subsequent calls using the token to request a protected resource are provided. Example sequence is provided.

Components

Static token

The token is a "static token" which doesn't perform any cryptography, just encodes data. The whole sequence, or a value contained within, is used for authentication.

The authentication client has a reader capable of reading the token.

The token could be an RFID card, a magnetic-stripe card, or anything else with enough information to be unique.

The static token's identifying information is hashed into an OAuth access token and a different OAuth refresh token.

OAuth tokens have a maximum of 500 characters, and are generated by passing the static token through the PBKDF2 algorithm with SHA-256, with a separate number of iterations.

ISAM API Protection (OAuth) Definition and Client

The OAuth definition encompasses all registrations and contains configuration for the grant lifetime, issuing and refreshing.

Two OAuth clients are created:

- client - used by the authentication library
- administration – support/security management

Additional client definitions can be created to partition additional login partitioning or authentication applications.

ISAM API Protection (OAuth) Mapping Rule

The mapping rules implement the business logic of the authentication and management flows, including:

- Enforcement of a maximum number of grants (registrations).
- Prevention of token reuse.
- Checking for an already-active grant and rejection of the request.
- Creation of a custom grant type to register a token.
- If the user self-care scenario includes token registration, then this grant will require username, password, plus the two hashed tokens.
- If the request comes from the administrator client, this grant omits the password.
- Extension of the "refresh_token" grant type to require a password.
- Custom "grant type" parameter once step-up is required, which looks up the token's owner and returns a username for the client to display in a native UI window.

The mapping rules are provided in the **isam** directory.

Authentication Library

The authentication library begins the authentication process and includes callback functions to handle the responses from the Mapping Rule business logic.

Details including the ISAM host, OAuth endpoint, client ID and client secret (for Administration) must be provided during initialization.

Refer to the README.md of the Git repository export in the **src** directory.

Lost and Stolen Tokens

In order for ISAM to revoke a static token indefinitely, it needs a grant tied to a user. When a token is to be indefinitely revoked, the value of the lost or stolen token is first removed from the registered user and assigned to the “locking user”. The mapping rule prevents duplicates tokens from being registered and prevents use of a revoked token.

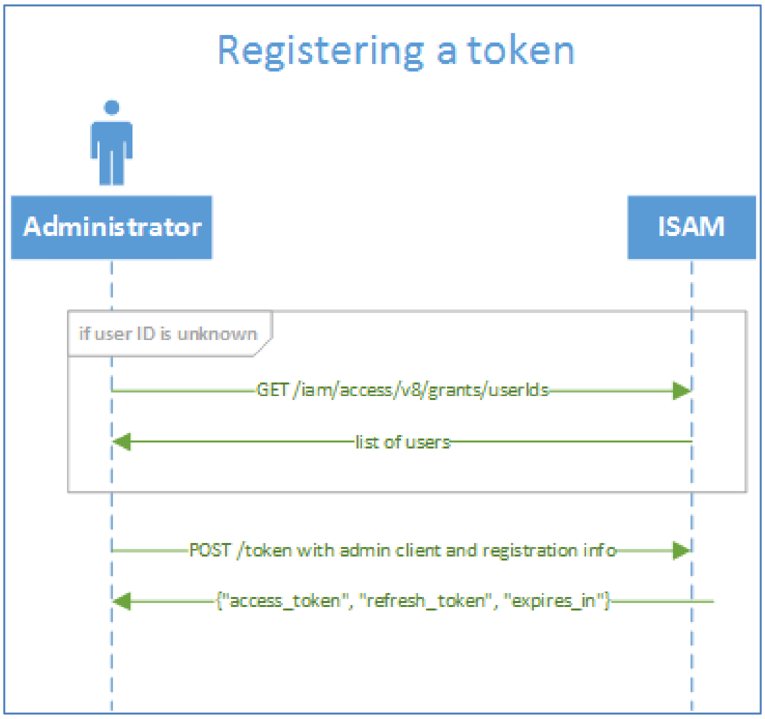
A “locking” user is created in ISAM's local user registry with a unique name, with *account-valid* and *password-valid* set to false which prevents the account from being used for interactive login or using the authentication library.

The account name should be unique and not used in any other registry. The account-valid setting prevents it from being used to access protected resources however must be created with a complex password.

Architectural Flows

Static token registration

Administrator

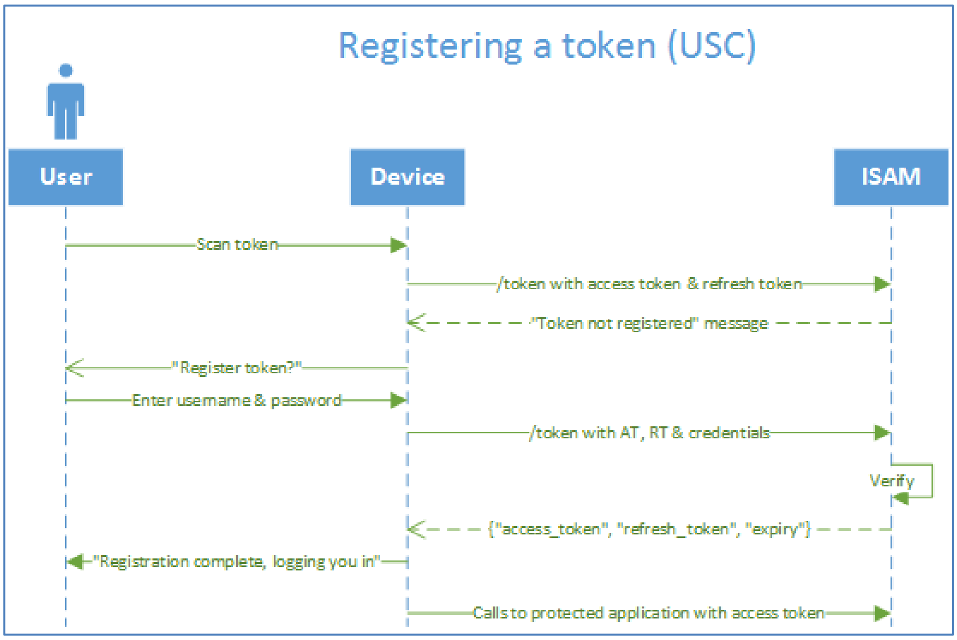

Registering a token
Administrator
ISAM
if user ID is unknown
GET /iam/access/v8/grants/userIds
list of users
POST /token with admin client and registration info
{"access_token", "refresh_token", "expires_in"}

1. (optional) If the username is unknown, the administrator retrieves a list of users.

2. The administrator enters the name of the user to register and scans the token. This generates the access token and refresh token used by ISAM, and sends it through.

3. ISAM returns a success message with those tokens.

User Self Care

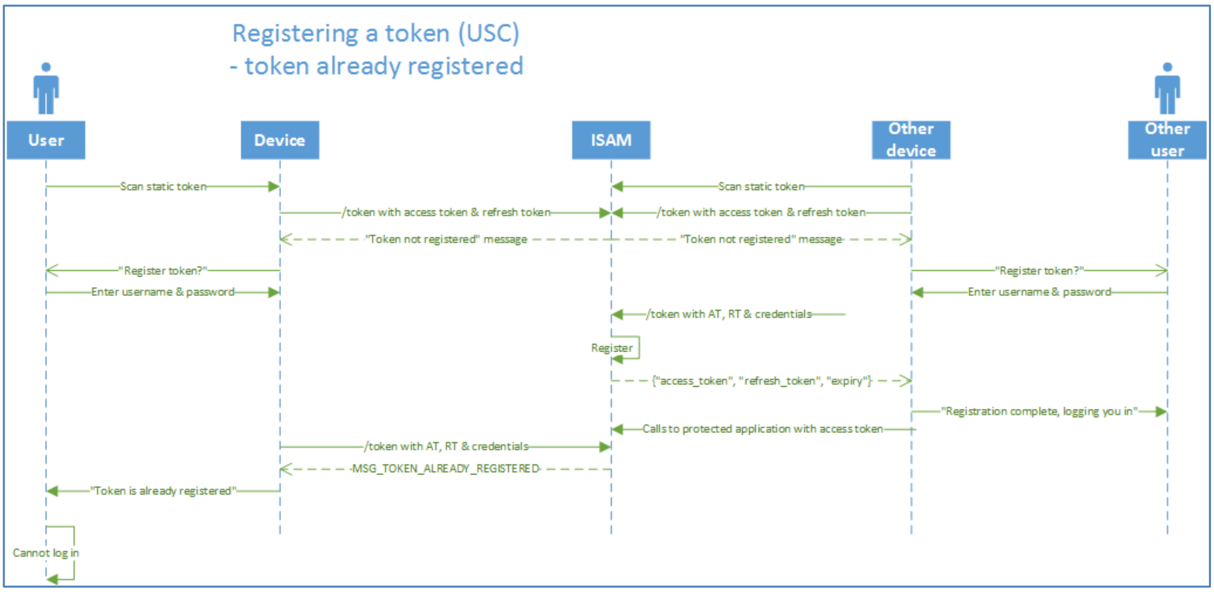

Registering a token (USC)
User
Device
ISAM
Scan token
/token with access token & refresh token
"Token not registered" message
"Register token?"
Enter username & password
/token with AT, RT & credentials
Verify
{"access_token", "refresh_token", "expiry"}
"Registration complete, logging you in"
Calls to protected application with access token

```
curl -kLs -XPOST -H 'Accept: application/json' -H 'Content-Type:
application/x-www-form-urlencoded'
"https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data
"username=testuser1&password=passw0rd&grant_type=register_token&clie
nt_id=THE_CLIENT&token_at=THE_ACCESS_TOKEN&token_rt=THE_REFRESH_TOKE
N"
```

```
{"access_token":"THE_ACCESS_TOKEN","refresh_token":"THE_REFRESH_TOKE
N","token_type":"bearer","expires_in":28799,"scope":""}
```

Token already registered

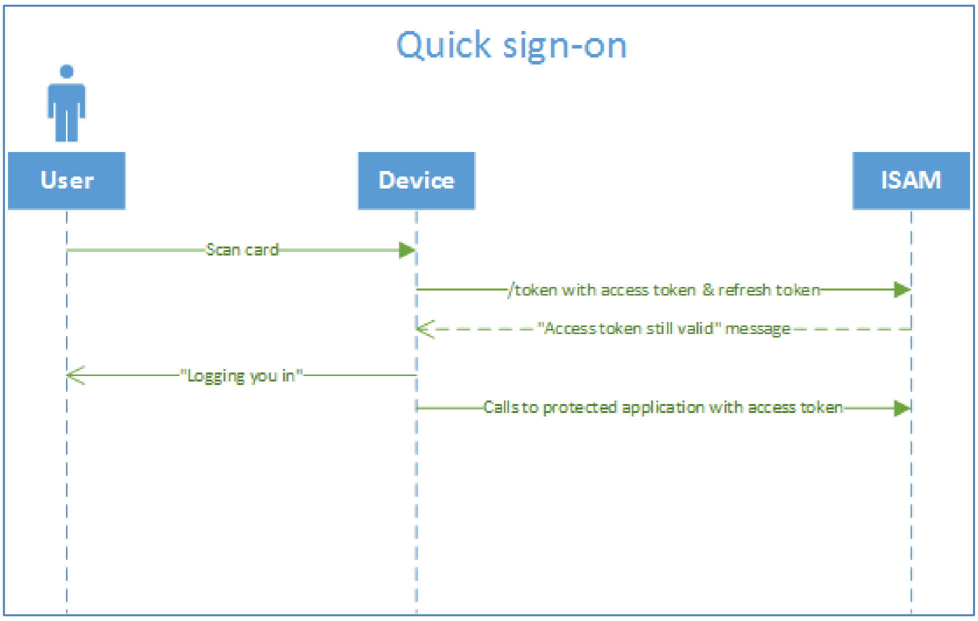
In a race condition, where the un-registered token is scanned twice by the same or different users, before one registers the static token:


Registering a token (USC)
- token already registered
User
Device
ISAM
Other device
Other user
Scan static token
Scan static token
/token with access token & refresh token
/token with access token & refresh token
"Token not registered" message
"Token not registered" message
"Register token?"
"Register token?"
Enter username & password
Enter username & password
/token with AT, RT & credentials
Register
{"access_token", "refresh_token", "expiry"}
"Registration complete, logging you in"
Calls to protected application with access token
/token with AT, RT & credentials
MSG_TOKEN_ALREADY_REGISTERED
"Token is already registered"
Cannot log in

```
curl -kLs -XPOST -H 'Accept: application/json' -H 'Content-Type:
application/x-www-form-urlencoded'
"https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data
"username=testuser1&password=passw0rd&grant_type=register_smartcard&
client_id=THE_CLIENT&token_at=THE_ACCESS_TOKEN&token_rt=THE_REFRESH_
TOKEN"
```

```
{"error":"mapping_error","error_description":"Tell the user this
token is already registered"}
```

“Day Pass” Quick sign-on


Quick sign-on
User
Device
ISAM
Scan card
/token with access token & refresh token
"Access token still valid" message
"Logging you in"
Calls to protected application with access token

1. The user scans the token onto their client device.

2. The device hashes the token into an access token & refresh token, and makes a refresh token request to ISAM (asking to verify that they're current).

3. ISAM returns a message that the access token can be used. Nothing has changed.

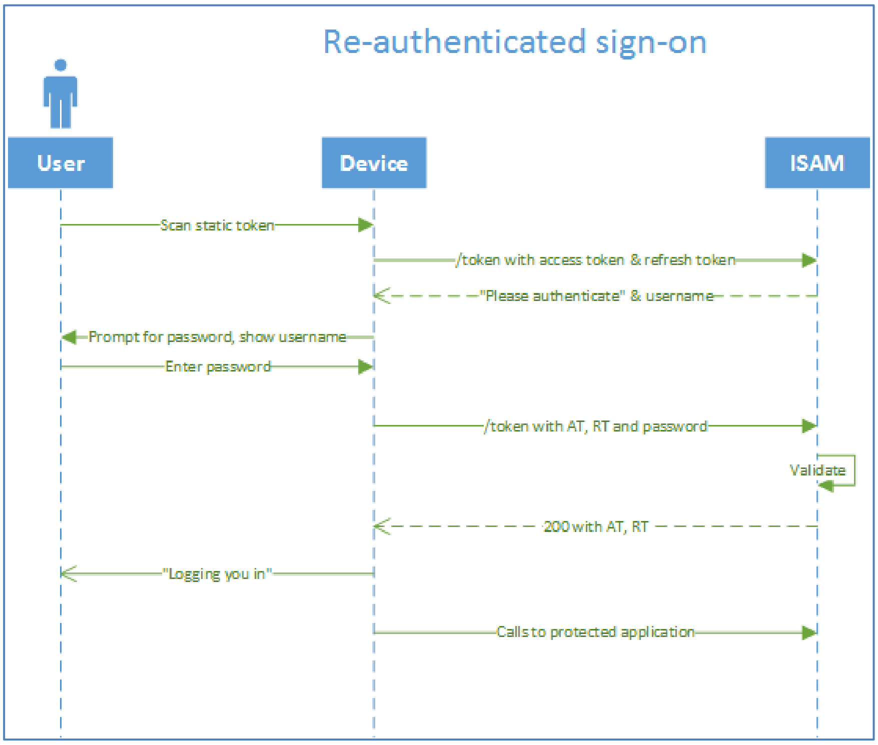
4. The device begins to log in the user, and uses the access token to access the protected application.

```
curl -kLs -XPOST -H 'Accept: application/json'
'https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token' --data
"grant_type=refresh_token&token_at=THE_ACCESS_TOKEN&refresh_token=TH
E_REFRESH_TOKEN&client_id=THE_CLIENT"
```

```
{"error":"mapping_error","error_description":"The grant is active
and you can log in"}
```

Re-authenticated sign-on

This applies if the optional "password challenge" step is enabled, and extends it with the optional username display.


Re-authenticated sign-on
User
Device
ISAM
Scan static token
/token with access token & refresh token
"Please authenticate" & username
Prompt for password, show username
Enter password
/token with AT, RT and password
Validate
200 with AT, RT
"Logging you in"
Calls to protected application

1. The user scans the token onto their client device.

2. The device hashes the token into an access token & refresh token, and makes a refresh token request to ISAM (asking to verify that they're current).

3. ISAM sends back the "you must authenticate" message and (optionally) a username to prompt with. The username is determined from the refresh token.

4. The device prompts the user for password login, optionally with the username.

5. The user enters their password.

6. The device sends another refresh token request, this time adding &password=<the user's entered password>, but not the username.

7. ISAM validates that the password matches the user, then sends a 200 response with the access token and refresh token.

8. The device begins to log in the user, and uses the access token to authenticate with the protected application behind ISAM.

```
curl -kLs -XPOST -H 'Accept: application/json'
'https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token' --data
"grant_type=refresh_token&token_at=THE_ACCESS_TOKEN&refresh_token=TH
E_REFRESH_TOKEN&client_id=THE_CLIENT"
```

```
{"error":"mapping_error","error_description":"Perform a password
refresh with user 'testuser1'"}
```

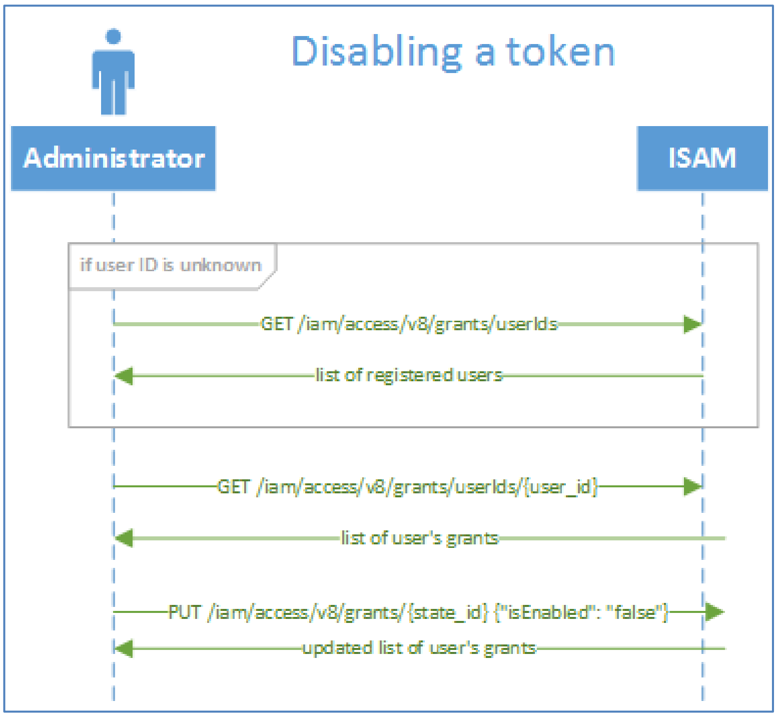
```
curl -kLs -XPOST -H 'Accept: application/json'
'https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token' --data
"grant_type=refresh_token&token_at=THE_ACCESS_TOKEN&refresh_token=TH
E_REFRESH_TOKEN&client_id=THE_CLIENT&password=passw0rd"
```

```
{"access_token":"THE_ACCESS_TOKEN","refresh_token":"THE_REFRESH_TOKE
N","token_type":"bearer","expires_in":0,"scope":""}
```

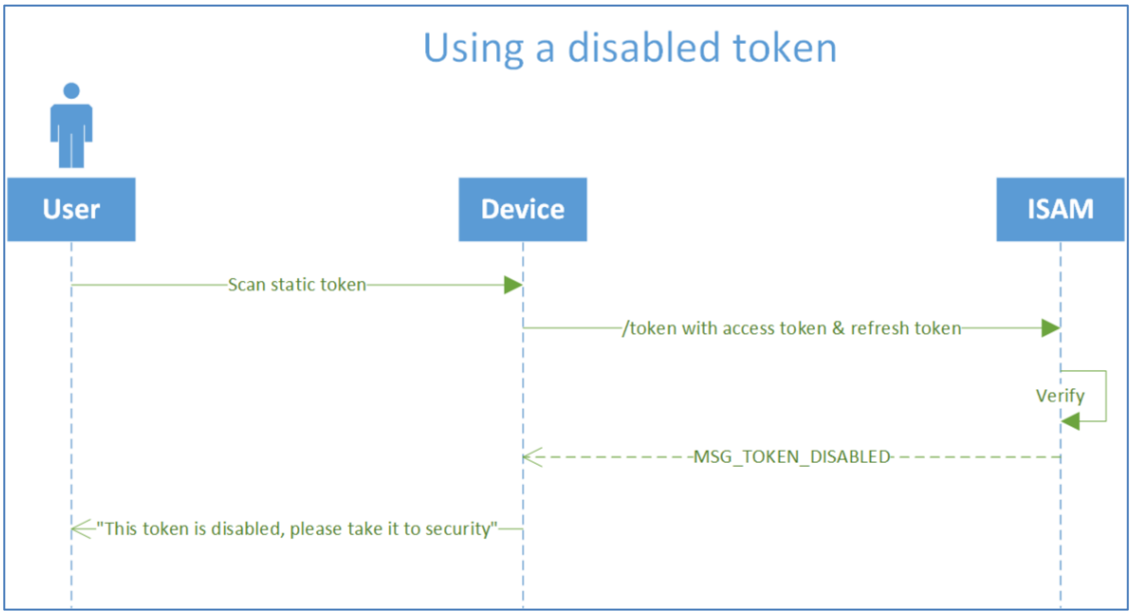
Static token revocation

Temporarily disable a token

The custom management page calls the RAPIs implements this:


Disabling a token
Administrator
ISAM
if user ID is unknown
GET /iam/access/v8/grants/userIds
list of registered users
GET /iam/access/v8/grants/userIds/{user_id}
list of user's grants
PUT /iam/access/v8/grants/{state_id} {"isEnabled": "false"}
updated list of user's grants

Upon attempted use

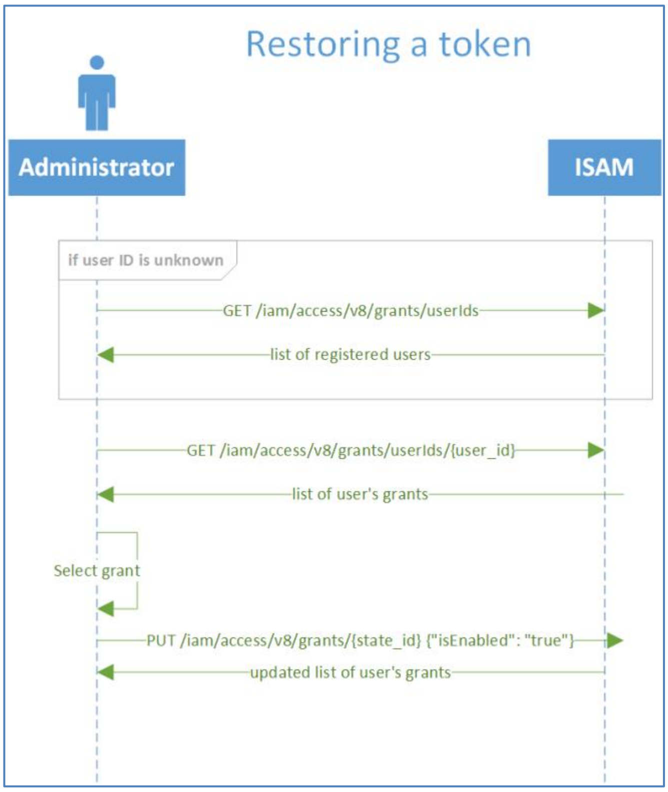

Using a disabled token
User
Device
ISAM
Scan static token
/token with access token & refresh token
Verify
MSG_TOKEN_DISABLED
"This token is disabled, please take it to security"

```
curl -kLs -XPOST -H 'Accept: application/json'
'https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token' --data
"grant_type=refresh_token&token_at=THE_ACCESS_TOKEN&refresh_token=TH
E_REFRESH_TOKEN&client_id=THE_CLIENT"
```

```
{"error":"mapping_error","error_description":"This ID card has been
locked."}
```

Restoring a revoked token

Restoring (Administrator)


Restoring a token
Administrator
ISAM
if user ID is unknown
GET /iam/access/v8/grants/userIds
list of registered users
GET /iam/access/v8/grants/userIds/{user_id}
list of user's grants
Select grant
PUT /iam/access/v8/grants/{state_id} {"isEnabled": "true"}
updated list of user's grants

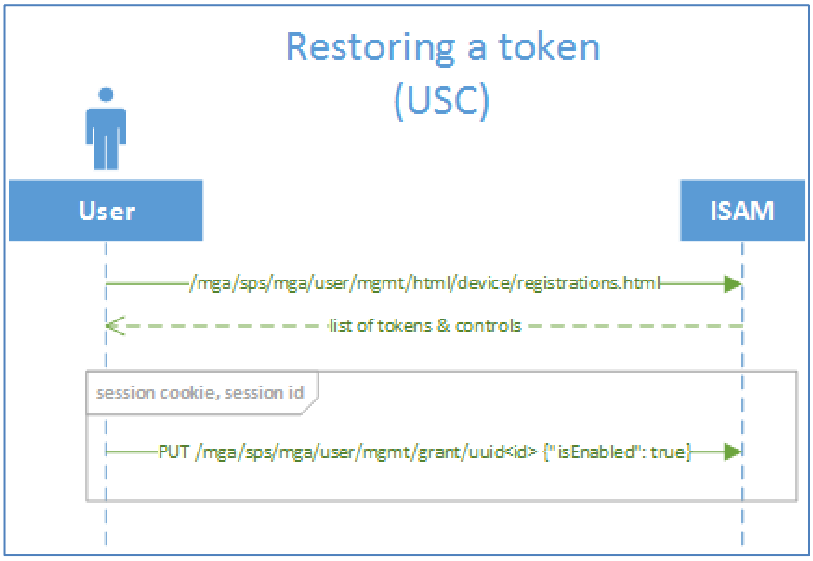
```
curl -kLs 'https://isam.myidp.ibm.com/iam/access/v8/grants' -u
admin:admin
```

```
[{"userId": "testuser1"}, {"userId": "steve"}]
```

```
curl -kLs
'https://isam.myidp.ibm.com/iam/access/v8/grants/userIds/steve' -u
admin:admin
```

```
[{"id":"uuid3180f4d4-015b-1a62-853b-
add2e515dc95","isEnabled":true,"clientId":"THE_CLIENT","tokens":[{"t
okenId":"THE_REFRESH_TOKEN","tokenString":"THE_REFRESH_TOKEN","type"
:"authorization_grant","subType":"refresh_token","dateCreated":"2017
-04-03T01:49:46Z","lifetime":60000,"lastUsed":"2017-04-
03T01:49:46Z","scope":""}],"attributes":[]}]
```

Restoring (User Self Care)


Restoring a token
(USC)
User
ISAM
/mga/sps/mga/user/mgmt/html/device/registrations.html
list of tokens & controls
session cookie, session id
PUT /mga/sps/mga/user/mgmt/grant/uuid<id> {"isEnabled": true}

```
curl -kLs -u testuser1:passw0rd -XPUT
'https://www.myidp.ibm.com/mga/sps/mga/user/mgmt/grant/uuid3bb2d629-
015b-1f27-b31d-add2e515dc95' --data '{"isEnabled": true}'
```

```
{"result":"FBTRBA199I The authorization grant uuid3bb2d629-015b-
1f27-b31d-add2e515dc95 was updated successfully."}
```

Users can self-manage their registered tokens to disable, enable or remove them.

1. Access the Manage Devices page via the Web Reverse proxy.
   For example:
   `https://www.myidp.ibm.com/mga/sps/mga/user/mgmt/html/device/device_selection.html`
2. Enter the credentials for the user.
3. Follow the prompts to register, disable or remove OAuth Grants.

IBM Security Access Manager Configuration

Complete the following configuration steps on the IBM Security Access appliance for integration with Memorial Hermann Authentication Plugin.

API Protection (OAuth) Configuration

Create an API Definition which will be used by the API Clients to configure the plugin. The default mapping rules will be updated with the contents of the mapping rules provided in the **isam** directory.

API Definition

Create a new API Definition using the Local Management Interface

1. Open the IBM Security Access Manager Local Management Interface
   For example:
   https://isam.myidp.ibm.com

2. Select **Secure Access Control → Policy → API Protection**

3. Select the **Definitions** menu item

4. Click the **Create Definition** (**+**) icon

5. Enter the **Name** for the definition.
   For example:
   Memorial Hermann Authentication Plugin

6. Optionally enter a **Description**
   For example:
   Static token to OAuth 2.0 mapping

7. Expand **Grant Types**

8. Uncheck **Authorization code**

9. Check **Resource owner username password**
   Note: This should be the only grant type enabled

10. Expand **Token management**

11. Specify the **Access token lifetime (seconds)** to the length of time a static token can be used to authentication before a password is required
    For example:
    28,800 requires re-authentication after 8 hours

12. **Access token length** is ignored as the mapping rule will create the token of the length corresponding to the static token value

13. Check **Issue refresh token**

14. Specify the **Maximum authorization grant lifetime (seconds**) to the duration of a registration at which point users will be required to reregister
    For example:
    31,536,000 is one year, but the scenario may advise five or ten

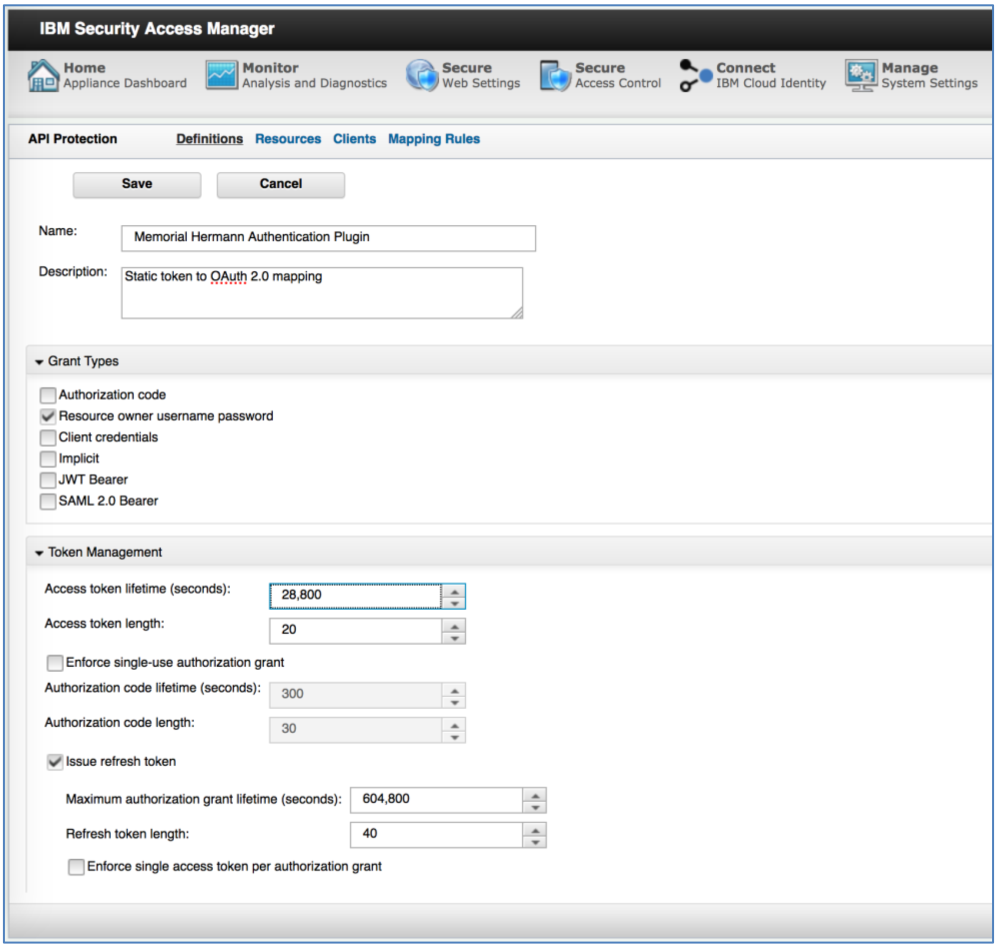
15. **Refresh token** length is ignored as the mapping rule will create the token of the length corresponding to the static token value

16. Uncheck **Enforce single access token per authorization grant**

17. Check **Enable multiple refresh tokens for fault tolerance**

18. Click **Save**

19. Review and Deploy the Pending Changes



Name: Memorial Hermann Authentication Plugin

Description: Static token to OAuth 2.0 mapping

Grant Types

- □ Authorization code
- ☑ Resource owner username password
- □ Client credentials
- □ Implicit
- □ JWT Bearer
- □ SAML 2.0 Bearer

Token Management

Access token lifetime (seconds): 28,800

Access token length: 20

□ Enforce single-use authorization grant

Authorization code lifetime (seconds): 300

Authorization code length: 30

☑ Issue refresh token

Maximum authorization grant lifetime (seconds): 604,800

Refresh token length: 40

□ Enforce single access token per authorization grant

API Client for User Authentication

Create a new API Client using the Local Management Interface

1. Open the IBM Security Access Manager Local Management Interface
   For example:
   https://isam.myidp.ibm.com

2. Select **Secure Access Control → Policy → API Protection**

3. Select the **Clients** menu item

4. Click the **New Client** (**+**) icon

5. Either accept the default generated **Client ID** or specify a value.
   This value must be provided during the initialization of the authentication and is used by the OAuth 2.0 client
   For example:
   OAuth20MemorialHermannUserAuth

6. Enter the **Client name**
   For example:
   Memorial Hermann Static Token User Authentication Client

7. Select the **API definition** corresponding to the definition created for this scenario
   For example:
   Memorial Hermann Authentication Plugin

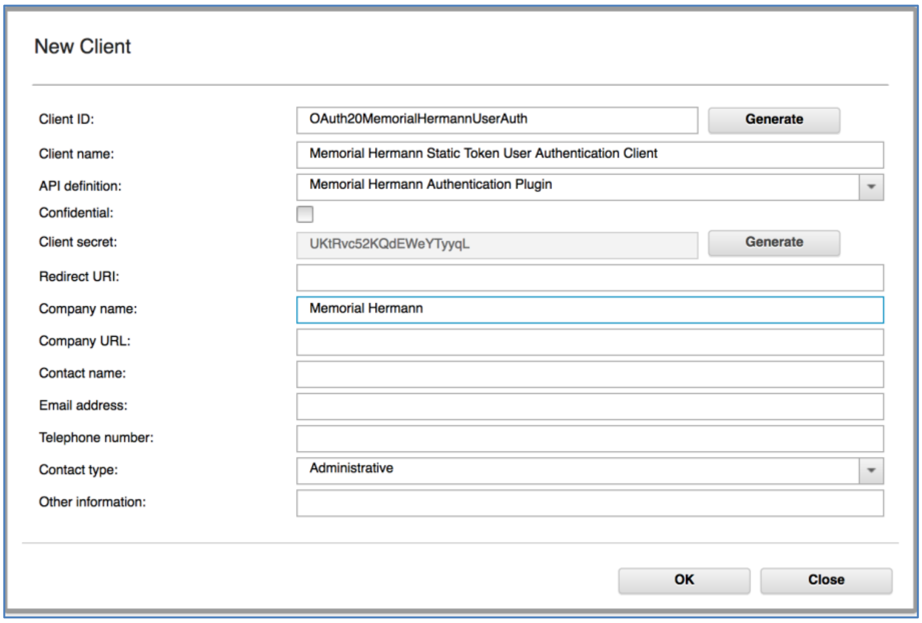
8. Uncheck **Confidential**

9. Enter the **Company name**
   For example:
   Memorial Hermann

10. Complete any of the additional optional fields

11. Click **OK**

12. Review and Deploy the Pending Changes



New Client

Client ID:	OAuth20MemorialHermannUserAuth
Client name:	Memorial Hermann Static Token User Authentication Client
API definition:	Memorial Hermann Authentication Plugin
Confidential:	□
Client secret:	UKtRvc52KQdEWeYTyyqL
Redirect URI:	
Company name:	Memorial Hermann
Company URL:	
Contact name:	
Email address:	
Telephone number:	
Contact type:	Administrative
Other information:	

API Client for User Management

Create a new API Client using the Local Management Interface

1. Open the IBM Security Access Manager Local Management Interface
   For example:
   https://isam.myidp.ibm.com

2. Select **Secure Access Control → Policy → API Protection**

3. Select the **Clients** menu item

4. Click the **New Client** (**+**) icon

5. Enter the **Client ID**.
   This value must be specified when calling the RAPIs or sample cURL commands for administration flows
   For example:
   Administrator

   Note: To use a different value you must update the line
   `var ADMINISTRATOR_CLIENT = "Administrator";`
   in the **PreTokenGeneration** mapping rule with the updated value

6. Enter the **Client name**
   For example:
   Memorial Hermann Static Token Administration Client

7. Select the **API definition** corresponding to the definition created for this scenario
   For example:
   Memorial Hermann Authentication Plugin

8. Check **Confidential**

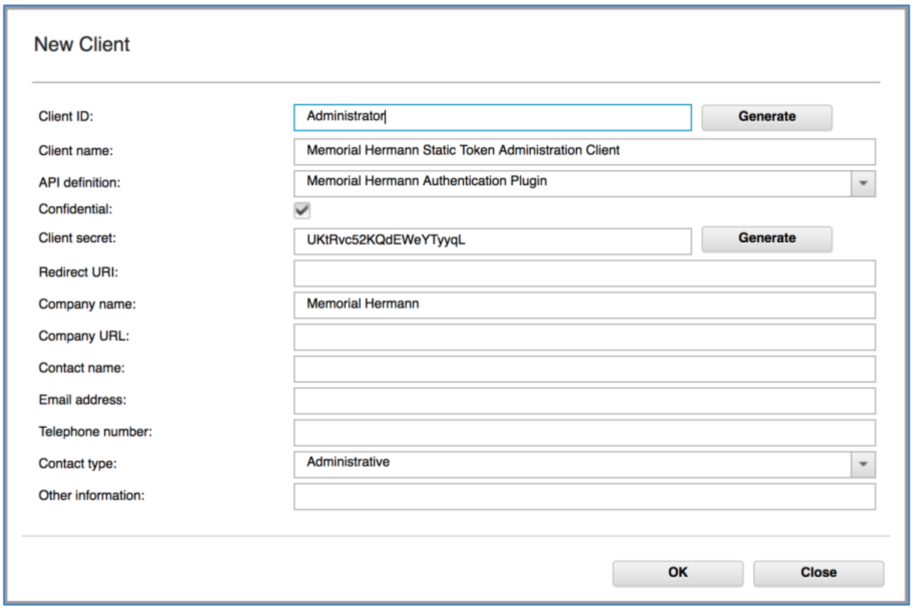
9. Either accept the default generated **Client secret** or specify a value. This value must be specified when calling the RAPIs or sample cURL commands for administration flows

10. Enter the **Company name**
    For example:
    Memorial Hermann

11. Complete any of the additional optional fields

12. Click **OK**

13. Review and Deploy the Pending Changes



New Client

Client ID:	Administrator
Client name:	Memorial Hermann Static Token Administration Client
API definition:	Memorial Hermann Authentication Plugin
Confidential:	☑
Client secret:	UKtRvc52KQdEWeYTyyqL
Redirect URI:	
Company name:	Memorial Hermann
Company URL:	
Contact name:	
Email address:	
Telephone number:	
Contact type:	Administrative
Other information:	

Replacing Default Mapping Rule Logic

The Mapping Rules contain the business logic and authentication flow customisations.

Use the Local Management Interface to replace the default mapping rules which were created by the API Definition.

1. Open the IBM Security Access Manager Local Management Interface
   For example:
   https://isam.myidp.ibm.com

2. Select **Secure Access Control → Policy → API Protection**

3. Select the **Mapping Rules** menu item

4. Select the PreTokenGeneration OAUTH mapping rule corresponding to the API Definition created
   For example:
   Memorial Hermann Authentication PluginPreTokenGeneration

5. Click the **Edit** icon

6. Select the entire contents in the editor window and remove the mapping rule JavaScript

7. Locate and open the *PreTokenGeneration.*js located in the **isam** directory

8. Copy the contents of *PreTokenGeneration.*js in to the blank editor window of the **Memorial Hermann Authentication PluginPreTokenGeneration** mapping rule

9. Click **Save**

10. Select the PostTokenGeneration OAUTH mapping rule corresponding to the API Definition created
    For example:
    Memorial Hermann Authentication PluginPostTokenGeneration

11. Click the **Edit** icon

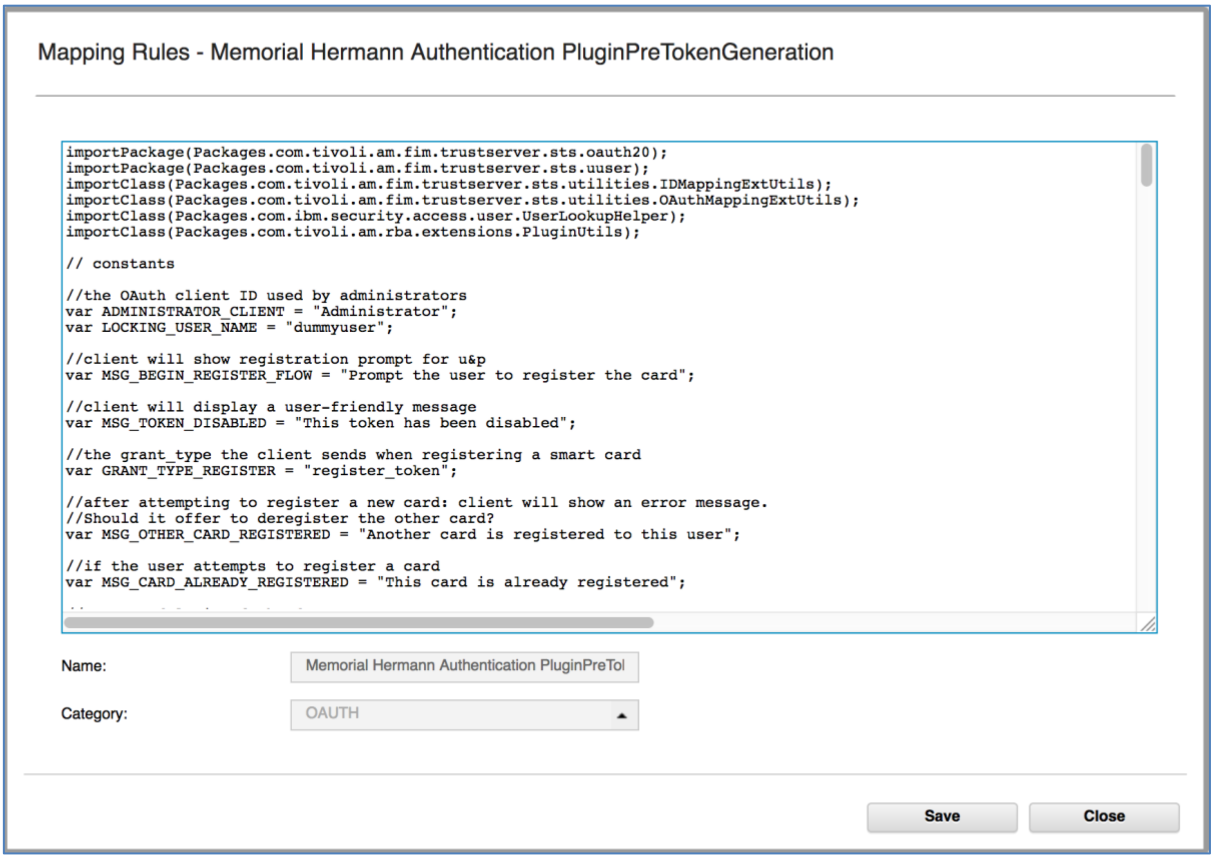
12. Select the entire contents in the editor window and remove the mapping rule JavaScript

13. Locate and open the *PostTokenGeneration.*js located in the **isam** directory

14. Copy the contents of *PostTokenGeneration.*js in to the blank editor window of the **Memorial Hermann Authentication PluginPostTokenGeneration** mapping rule

15. Click **Save**

16. Review and Deploy the pending changes



Mapping Rules - Memorial Hermann Authentication PluginPreTokenGeneration

```
importPackage(Packages.com.tivoli.am.fim.trustserver.sts.oauth20);
importPackage(Packages.com.tivoli.am.fim.trustserver.sts.uuser);
importClass(Packages.com.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils);
importClass(Packages.com.tivoli.am.fim.trustserver.sts.utilities.OAuthMappingExtUtils);
importClass(Packages.com.ibm.security.access.user.UserLookupHelper);
importClass(Packages.com.tivoli.am.rba.extensions.PluginUtils);

// constants

//the OAuth client ID used by administrators
var ADMINISTRATOR_CLIENT = "Administrator";
var LOCKING_USER_NAME = "dummyuser";

//client will show registration prompt for u&p
var MSG_BEGIN_REGISTER_FLOW = "Prompt the user to register the card";

//client will display a user-friendly message
var MSG_TOKEN_DISABLED = "This token has been disabled";

//the grant_type the client sends when registering a smart card
var GRANT_TYPE_REGISTER = "register_token";

//after attempting to register a new card: client will show an error message.
//Should it offer to deregister the other card?
var MSG_OTHER_CARD_REGISTERED = "Another card is registered to this user";

//if the user attempts to register a card
var MSG_CARD_ALREADY_REGISTERED = "This card is already registered";
```

Name: Memorial Hermann Authentication PluginPreTol

Category: OAUTH

Configuring Password Authentication Mechanism

In order for the OAuth mapping rule to validate the username and password for token registration, or to validate the password during the authentication flow after the access token has reached its maximum lifetime, the Advanced Access Control Password Authentication Mechanism must be configured.

1. Open the IBM Security Access Manager Local Management Interface.
   For example:
   https://isam.myidp.ibm.com

2. Select **Secure Access Control → Policy → Authentication**

3. Select the **Mechanisms** menu item

4. Select the **Username Password** mechanism

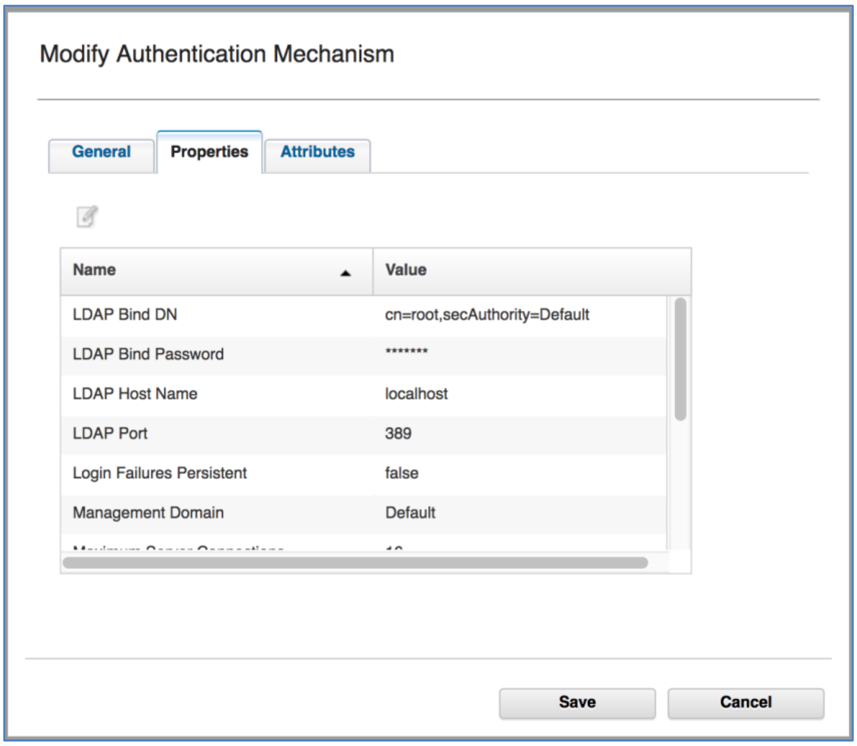
5. Click the **Modify Authentication Mechanism** icon

6. Select the **Properties** tab

7. Update the configuration values for the User Directory configured for ISAM.
   This may be an external supported directory server or the on-box LDAP.
   For example:
   Select the required row and click the Modify icon for each item.
   Update the required Value and click OK to save that change
   a. Set **LDAP Bind DN** as cn=root,secAuthority=default
   b. Set **LDAP Bind Password** as passw0rd
   c. Set **LDAP Host Name** as localhost
   d. Set **LDAP Port** as 389
   e. Set **SSL Enabled** as False (unchecked)
   f. If using a Federated Directory in the ISAM Runtime Configuration set **Use Federated Directories Configuration** as True (checked)

8. Click **Save**

9. Review and Deploy the pending changes



Modify Authentication Mechanism

Name	Value
LDAP Bind DN	cn=root,secAuthority=Default
LDAP Bind Password	*******
LDAP Host Name	localhost
LDAP Port	389
Login Failures Persistent	false
Management Domain	Default
Maximum Server Connections	16

Locking User account creation

When a token is reported lost or stolen and is to be permanently revoked to prevent reuse, the value of the token (the grant) is removed from the user it was registered to and assigned to the locking user account and then disabled.
This prevents the card from being used or reregistered.
Ensure the password is of sufficient length and complexity.

1. Use **Policy Administration** or the **`pdadmin`** command-line utility to create the locking user account to bind permanently revoked tokens to.
   For example:
   ```
   pdadmin sec_master> user create dummyuser uid=dummyuser,dc=iswga
   Locked User <random_and_complex_password>
   pdadmin sec_master> user modify dummyuser account-valid no
   pdadmin sec_master> user modify dummyuser password-valid no
   ```

Note: To use a different account you must update the line
`var LOCKING_USER_NAME = "dummyuser";`
in the **PreTokenGeneration** mapping rule with the updated value and specify the account when performing the grant creation with the value of the lost or stolen token.

ESSO IMS Integration

The Authentication Plugin is intended to enable the use of a Static Token such as physical access card credential to authenticate to ISAM in order to retrieve the user's real desktop credentials.

Deployment of ESSO and the Single Sign On between ESSO and ISAM is not a component of this integration, however high-level detail is provided to complete and end to end scenario.

ESSO Web API Installation and Configuration

Follow the instructions in the Knowledge Center to enable the Web API for IBM Security Access Manager for Enterprise Single Sign-On, Version 8.2.2.

Refer to https://www.ibm.com/support/knowledgecenter/en/SS9JLE_8.2.2/com.ibm.itamesso.doc_8.2.2/WebAPI/concepts/web_api_intro.html

Note: IBM Security Access Manager for Enterprise Single Sign-On, Version 8.2.1 requires the deployment of IBM Tivoli Federated Identity Manager Security Token Service (STS) and is not recommended. Refer to http://www-01.ibm.com/support/docview.wss?uid=swg21691362

ESSO IMS Single Sign On

Once the Web API has been installed and configured, the ESSO RAPIs can be called to retrieve the system credentials for the authenticated user. When calling the RAPIs directly, the user account that contains the managed credentials is used to authenticate to IMS.

Configure Single Sign On from ISAM to ESSO to enable retrieval of user credentials once the ISAM user has authenticated with their Static Token.

User Account Synchronization

The ESSO IMS user repository configured in the WebSphere Application Server should be configured in the ISAM appliance to achieve SSO which ensures the accounts are synchronized.

Follow the ISAM Knowledge Center for Managing federated directories and Configuring the runtime to authenticate basic users.

Refer to https://www.ibm.com/support/knowledgecenter/en/SSPREK_9.0.3/com.ibm.isam.doc/admin/concept/con_usersanduserregistries.html

Single Sign On mechanism from ISAM to ESSO

Multiple mechanisms exist to configure SSO from ISAM to ESSO and the ESSO RAPIs, including

- LTPA – recommended, low complexity allow direct SSO from the provided HTTP request cookie
- SAML 2.0 TAI – alternative to LTPA allowing direct SSO from a provided HTTP request header
- eTAI – not recommended

The method implemented is a deployment consideration and will be dependent on many factors including

- currently deployed infrastructure
- licenced product offerings (for example ISAM 9 Federation)
- existing SSO from ISAM to ESSO or WebSphere infrastructure
- deployment, scalability and management

Ensure a suitable SSO mechanism is implemented to ensure the identity of the user authenticating to ISAM is single signed on to the ESSO IMS to ensure the correct user credentials can be retrieved via the RAPIs.

Junction Creation

Retrieval of user credentials via the ESSO IMS RAPIs requires a single junction created on the ISAM Web Reverse Proxy.

The Web API is typically served from the `esso` context root.

For example:

```
https://<esso_ims_host>/esso/webapi/wallet/accounts
```

Create a Transparent Path junction to the same context root.

The **pdadmin** command-line utility can be used on the IBM Security Access Manager Appliance in addition to the graphical user interface of the Local Management Interface (LMI) for some of the integration steps.

For example, using pdadmin:

```
pdadmin sec_master> server task default-<instance>-<host>
create -t ssl -h <esso_ims_host> -p 443 -x -c
iv_user,iv_creds -F <ltpa_key_file> -Z
<ltpa_key_password> -f /esso
```

Obtaining User Credentials

IBM Security Access Manager for Enterprise Single Sign-On does not provide an SDK to call the RAPIs for account retrieve.

The logic to call the ESSO EMS RAPIs and parse the response for the user credentials should be authored into the calling application after the Authentication Plugin library has been called.

Refer to sample VBS script and RAPI documentation provided in the `esso` directory.

This sample authenticates to ISAM using the Basic username and password of the user account and can be used to validate the SSO between ISAM and ESSO. Ensure Basic authentication is enabled in ISAM to use this sample.

Refer to
https://www.ibm.com/support/knowledgecenter/en/SSPREK_9.0.3/com.ibm.isam.doc/wrp_config/concept/con_basic_authe.html

Once SSO has been validated and the Static Token for authentication has been configured authentication of the user can be completed using `oauth-auth` to retrieve the user credentials.

Refer to
https://www.ibm.com/support/knowledgecenter/en/SSPREK_9.0.3/com.ibm.isam.doc/wrp_config/concept/con_open_authentication_webseal.html

For example, using cURL

```
curl -k -c auth.txt -H "Authorization: Bearer
at_<static_token_value>"
https://<webseal_host>/esso/webapi/wallet/accounts
```

Note: The HTTP header provided, named "Authorization". The value of this header is the key word Bearer followed by the combining the prefix **`at_`** and the value read from the static token such as the physical credential that the user has swiped/tagged.

This access token is evaluated by the OAuth server and will allow the HTTP request to be satisfied for the protected resource, being the ESSO Web API.

Running the Sample Application

Use the example scenario to validate configuration of the custom OAuth 2.0 authentication plugin and library.

Before you start

This section does not cover the configuration of the entire environment. It focuses on running the test scenarios and performing the configuration with sample values.

References are made to the configuration steps provided in

- *IBM Security Access Manager Configuration* on page 24

User Account Creation

Ensure a valid user account is available in the configured ISAM user registry.

Validating Authentication

The **isamAuth** binaries included in the `sample` directory can be used to simulate the input value read from a physical credential to validate the library and mapping rules.

The sample application includes:

- Static token registration, requiring a username and password
- Fast pass, requiring only the sample Static Token card value
- Daily reauthentication, requiring only the user’s password

Sample Application Syntax

```
isamAuth(.exe) -h <ISAM hostname/IP> -p <port> -e <endpoint> -c
<client id> -s <isSecure> -t <token_string>
```

Parameter	Description
`-h (host)`	IBM Security Access Manager Web Reverse Proxy, including protocol. For example: *https://www.myidp.ibm.com*
`-p (port)`	IBM Security Access Manager Web Reverse

Parameter	Description
	Proxy port. For example: *443* for HTTPS
`-e (endpoint)`	IBM Security Access Manager Advanced Access Control junction path. For example: */mga*
`-c (client_id)`	IBM Security Access Manager Advanced Access Control OAuth 2.0 client_id For example: *OAuth20MemorialHermannUserAuth*
`-s (isSecure)`	Use SSL/TLS encryption on the connection (HTTP or HTTPS) Acceptable values: *0* – false *1* – true
`-t (Token String)`	Sample Static Token value to simulate that read from the physical user credential such as an RFID credential card. For example: *3497796977*

- *Table 2: isamAuth sample application syntax.*

Windows

Before executing the sample, ensure

- libcurl.dll is available
- Microsoft Visual C++ Runtime is installed

To execute the sample on Windows:

1. Copy libcurl.dll to the same directory as `isamAuth.exe`
2. Execute isamAuth.exe

   For example:
   ```
   C:\Users\user\isammhauth>isamAuth.exe -h
   https://www.myidp.ibm.com -p 443 -e /mga -s 1 -c
   OAuth20MemorialHermannUserAuth -t 3497796977
   ```

Linux

Before executing the sample, ensure

- libcurl.so available
- glibc-2.14 or above is installed

To execute the sample on Linux:

1. Add libcurl.so to the LD_LIBRARY_PATH if not already
   For example:
   ```
   export LD_LIBRARY_PATH=/home/user/isammhauth/sample
   ```
2. Execute isamAuth

   For example:
   ```
   [user@host sample]$ ./isamAuth -h https://www.myidp.ibm.com -p
   443 -e /mga -s 1 -c OAuth20MemorialHermannUserAuth -t
   3497796977
   ```

Sample Output

Review the sample output to demonstrate the registration, “day pass” quick sign on and reauthentication scenarios.

Registration

```
asPrintf: tempString is -h

Argv[1] is -h

Key is -h

asPrintf: tempString is https://www.myidp.ibm.com

asPrintf: tempString is -p

Argv[3] is -p

Key is -p

asPrintf: tempString is 443

asPrintf: tempString is -e

Argv[5] is -e

Key is -e

asPrintf: tempString is /mga

asPrintf: tempString is -c

Argv[7] is -c

Key is -c

asPrintf: tempString is OAuth20MemorialHermannUserAuth

asPrintf: tempString is -s
```

```
Argv[9] is -s

Key is -s

asPrintf: tempString is 1

asPrintf: tempString is -t

Argv[11] is -t

Key is -t

asPrintf: tempString is 3497796977

The value of i is 13.

Final endpoint URL is
https://www.myidp.ibm.com:443/mga/sps/oauth/oauth20/token.


Trying to get oauth token results in
{"error":"mapping_error","error_description":"Prompt the user to
register the card"}

ERROR:: (null)


Enter your username: testuser

Enter your password: Passw0rd1


Trying to get oauth token results in
{"access_token":"at_3497796977","refresh_token":"rt_3497796977","toke
n_type":"bearer","expires_in":28799,"scope":""}

Status: User successfully authenticated.
```

“Day Pass” Quick sign-on

```
asPrintf: tempString is -h

Argv[1] is -h

Key is -h

asPrintf: tempString is https://www.myidp.ibm.com

asPrintf: tempString is -p

Argv[3] is -p

Key is -p

asPrintf: tempString is 443

asPrintf: tempString is -e

Argv[5] is -e
```

```
Key is -e

asPrintf: tempString is /mga

asPrintf: tempString is -c

Argv[7] is -c

Key is -c

asPrintf: tempString is OAuth20MemorialHermannUserAuth

asPrintf: tempString is -s

Argv[9] is -s

Key is -s

asPrintf: tempString is 1

asPrintf: tempString is -t

Argv[11] is -t

Key is -t

asPrintf: tempString is 3497796977

The value of i is 13.

Final endpoint URL is
https://www.myidp.ibm.com:443/mga/sps/oauth/oauth20/token.

Trying to get oauth token results in
{"error":"mapping_error","error_description":"The grant is active and
you can log in"}

ERROR:: (null)


Status: User successfully authenticated.
```

Password Reauthentication

```
asPrintf: tempString is -h

Argv[1] is -h

Key is -h

asPrintf: tempString is https://www.myidp.ibm.com

asPrintf: tempString is -p

Argv[3] is -p

Key is -p

asPrintf: tempString is 443

asPrintf: tempString is -e
```

```
Argv[5] is -e
Key is -e
asPrintf: tempString is /mga
asPrintf: tempString is -c
Argv[7] is -c
Key is -c
asPrintf: tempString is OAuth20MemorialHermannUserAuth
asPrintf: tempString is -s
Argv[9] is -s
Key is -s
asPrintf: tempString is 1
asPrintf: tempString is -t
Argv[11] is -t
Key is -t
asPrintf: tempString is 3497796977
The value of i is 13.
Final endpoint URL is
https://www.myidp.ibm.com:443/mga/sps/oauth/oauth20/token.
Trying to get oauth token results in
{"error":"mapping_error","error_description":"Perform a password
refresh with user 'testuser'"}
ERROR:: (null)


Username is testuser
Enter the password for the user testuser: Passw0rd1


Trying to get oauth token results in
{"access_token":"at_3497796977","refresh_token":"rt_3497796977","toke
n_type":"bearer","expires_in":28799,"scope":""}
Status: User successfully authenticated.
```

Managing Tokens

Users can self-manage their registered tokens to disable, enable or remove them.

4. Access the Manage Devices page via the Web Reverse proxy.
   For example:

```
https://www.myidp.ibm.com/mga/sps/mga/user/mgmt/html/device/de
vice_selection.html
```

5. Enter the credentials for the user.
6. Follow the prompts to register, disable or remove OAuth Grants.

You can also use the following URL to go directly to the attributes page of a registered token.
For example:

```
https://www.myidp.ibm.com/mga/sps/mga/user/mgmt/html/device/devic
e_attributes.html?id=3497796977.
```

The query string, ?id= indicates the device that you are trying to access.

Note: This template page can be modified to organisational theming and standards and for example, to remove the ability to remove gants.

Refer to
https://www.ibm.com/support/knowledgecenter/SSPREK_9.0.3/com.ibm.isam.doc/admin/task/tsk_lmi_modify_template_pages.html

cURL Command Reference

The architectural flows of the scenario can be exercised using cURL samples against the ISAM appliance to validate configuration values and customisation instead of the sample appliance.

User Scenarios

Registration

```
curl -kLs -XPOST -H 'Accept: application/json' -H 'Content-Type:
application/x-www-form-urlencoded'
"https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data
"username=testuser1&password=passw0rd&grant_type=register_smartcard&
client_id=OAuth20MemorialHermannUserAuth&smartcard_at=at_${smartcard
}&smartcard_rt=rt_${smartcard}"
```

Receive the usual access_token & refresh_token.

Note that grant_type = register_smartcard.

Authentication

To begin grant refresh & get a username (at start of shift):

```
curl -kLs -XPOST -H 'Accept: application/json'
'https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token' --data
"grant_type=refresh_token&client_id=OAuth20MemorialHermannUserAuth&r
efresh_token=rt_${smartcard}&smartcard_at=at_${smartcard}"
```

Receive a mapping error with error_description “Perform a password refresh with user ‘testuser1’” OR a mapping error with error_description “Prompt the user to register the card“ OR a mapping error with error_description “The grant is active and you can log in”.

Reauthentication

```
curl -kLs -XPOST -H 'Accept: application/json'
'https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token' --data
"grant_type=refresh_token&refresh_token=rt_${smartcard}&smartcard_at
=at_${smartcard}&client_id=OAuth20MemorialHermannUserAuth&password=p
assw0rd"
```

(this adds &password=passw0rd to the "begin refresh" flow)

Receive the usual access_token & refresh_token.

Administrator Scenarios

Retrieve all registered IDs

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/
```

Retrieve all registered users

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/userIds
```

Retrieve single user

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/userIds/{user_id}
```

Register Token

```
curl -kL -XPOST -H 'Accept: application/json' -H 'Content-Type:
application/x-www-form-urlencoded'
"https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data
"username=testuser1&grant_type=register_smartcard&client_id=Administ
rator&smartcard_at=at_${smartcard}&smartcard_rt=rt_${smartcard}&clie
nt_secret={oauth_client_secret_password}"
```

Disable Token

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/{state_id} -XPUT –
data ‘

{“isEnabled”: “false”}

‘
```

Enable Token

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/{state_id} -XPUT –
data ‘

{“isEnabled”: “true”}

‘
```

Permanently Lock a Token (Lost or Stolen)

Use this process to assign the value of the lost token to a dedicated account in ISAM that maintains lost and stolen grants so they cannot be reactivated in the future.

There is no management interface provided in ISAM to perform these steps and these RAPIs should be incorporated in to an existing administrator/help desk/security officer portal or application.

1. Determine the grant for the user reporting the card lost or stolen

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/userIds/{user_id}
```

Record the **id** in the JSON payload of the response

2. Remove the grants for the user
   Note: This will remove all grants and assumes there will be a single grant only. If additional grants are registered delete the specific id. Refer to the ISAM RAPIs for details.

```
curl -kL -XDELETE -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/userIds/{user_id}
```

3. Use the Administrator client_id to register the token to the dedicated lock account (dummyuser), passing the original lost/stolen id as the Access and Refresh token values.

```
curl -kL -XPOST -H 'Accept: application/json' -H 'Content-Type:
application/x-www-form-urlencoded'
"https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data
"username=dummyuser&grant_type=register_smartcard&client_id=Admin
istrator&smartcard_at=at_${smartcard}&smartcard_rt=rt_${smartcard
}&client_secret={oauth_client_secret_password}"
```

4. Disable the token using the original lost/stolen id.

```
curl -kL -u admin:admin
https://isam.myidp.ibm.com/iam/access/v8/grants/{state_id} -XPUT
—data ‵

{“isEnabled”: “false”}

‵
```

Error Message Reference

Register with the wrong password

```
curl -kL -XPOST -H 'Accept: application/json' -H 'Content-Type:
application/x-www-form-urlencoded'
"https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data
```

```
"username=testuser1&password=WRONG&grant_type=register_smartcard&client_id=MemorialHerman&smartcard_at=at_${smartcard}&smartcard_rt=rt_${smartcard}"
```

Receive a mapping error with error_description

- "HPDAA0329E   The credentials provided cannot be authenticated by the registry."

Attempt to register with a card already registered

```
curl -kL -XPOST -H 'Accept: application/json' -H 'Content-Type: application/x-www-form-urlencoded' "https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data "username=testuser1&password=passw0rd&grant_type=register_smartcard&client_id=MemorialHerman&smartcard_at=at_${smartcard}&smartcard_rt=rt_${smartcard}"
```

```
curl -kL -XPOST -H 'Accept: application/json' -H 'Content-Type: application/x-www-form-urlencoded' "https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data "username=steve&password=passw0rd&grant_type=register_smartcard&client_id=MemorialHerman&smartcard_at=at_${smartcard}&smartcard_rt=rt_${smartcard}"
```

Receive a mapping error with error_description

- {"error":"mapping_error","error_description":"Tell the user this card is already registered”}

Register a new card when you already have a registered card

```
curl -kL -XPOST -H 'Accept: application/json' -H 'Content-Type: application/x-www-form-urlencoded' "https://www.myidp.ibm.com/mga/sps/oauth/oauth20/token" --data "username=testuser1&password=passw0rd&grant_type=register_smartcard&client_id=MemorialHerman&smartcard_at=at_${smartcard}&smartcard_rt=rt_${smartcard}"
```

```
[repeat the command]
```

Receive a mapping error with error_description

- {“error":"mapping_error","error_description":"Tell the user another card is registered to them"}

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed 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 may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give 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 character set (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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 for the purpose of enabling: (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
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may 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 may vary significantly. Some measurements may 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 measurement may have been estimated through extrapolation. Actual results may 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 IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 conforming 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. 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 conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2017. Portions of this code are derived from IBM Corp. Sample Programs. ©Copyright IBM Corp 2017, 2017. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information; at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other company, product, and service names may be trademarks or service marks of others.