IBM Security

IBM Security Identity Governance and Intelligence

Version 5.2.5

Reverse Password Sync plug-in

Custom development

Table of Contents

Tables

Figures

Preface

This document is provided as a guide, along with the *ReversePwdSyncTutorial.zip* package, to describe the process of developing a custom *Reverse Password Sync* plug-in for the Identity Governance and Intelligence product (IGI) product.

The *ReversePwdSyncTutorial.zip* package contains a java project that demonstrate how to connect and propagate a password change from a target system to the Identity Governance and Intelligence using the published REST APIs. Note that each target system provides a different method to capture a password change, therefore capturing the password change on the target system will not be covered in this document and tutorial.

Overview

The purpose of the *Reverse Password Sync*plug-in in IGI is to maintain *password synchronization* for all accounts owned by a user even when the user changes his or her password on the target system. This implies that the *Password Sync*feature in IGI must be configured by creating a password synch group of the targets (applications) that will share the same password. The target where the *Reverse Password Sync*plug-in is deployed, must be managed by IGI and must be in the same password synch group of the targets that will share the same password. For more information on IGI password synchronization, refer to the Password Synchronization section and Table 1 - Password Synchronization Setup

Figure 1 - Active Directory Reverse Password Synch shows an example of how the Active Directory Reverse Password Plug-in deployment.

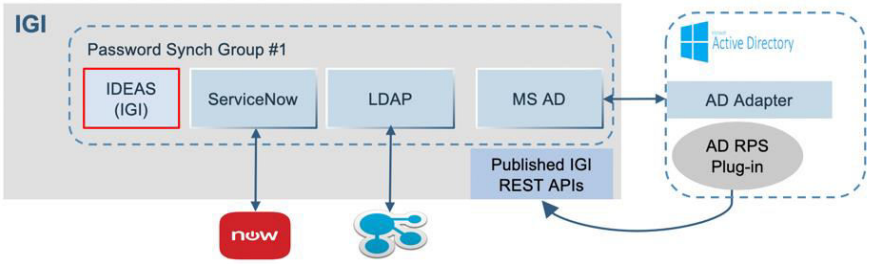


Figure 1 - Active Directory Reverse Password Synch

- The Active Directory domain is managed by IGI (MS AD).
- The Active Directory reverse password plug-in (AD RPS) is deployed on the target.
  - *Note: The AD RPS plug-in is supported out of the box by IGI and available for download.*
- On IGI, a password synch group is created that includes the MS AD and other applications.

Password Synchronization

Password synchronization is the process through which a user maintains a single password across multiple applications. See Table 1 - Password Synchronization Setup

Identity Governance and Intelligence implementation

- Ability to select the applications that will participate in password synchronization. This is referred to as a *Password Synch Group*
- Specific password policy per Password Synch Group
- Ability to create multiple Password Synch Groups
- Optionally, you can include the IGI account configuration into a password synch group

Setup tasks	Topics
1. Add and define a password sync group.	• Adding password sync groups • Password Sync Configurations
2. Define a password policy for a password sync group.	• Defining a password policy for a password sync group • Password Policy for Password Sync Groups
3. Add account configurations for the target to a password sync group.	• Adding or removing account configurations for password sync groups • Account Configurations for Password Sync Groups

Table 1 - Password Synchronization Setup

Reverse Password Synchronization

Reverse password synchronization is the process where a password change on one of the target systems or applications, such as Windows Active Directory, is used to synchronize all of the other account passwords for a user within the same password sync group.

- The target system or application where the password is changed and synchronized must be managed by Identity Governance and Intelligence.
- The plug-in must be deployed on the target system where the password will be synchronized.
- The plug-in notifies Identity Governance and Intelligence of a password change via the standard published Identity Governance and Intelligence REST APIs by using secure communication.
- All of the user's accounts that are within the same password sync group are affected.
- Built-in recursion detection: Identity Governance and Intelligence will not issue a password change for the same target system where the synchronized password change was originated.

Reverse Password Sync Plug-in Logic and Dataflow

Most targets provide a method that will enable you to deploy custom code, plug-in, that will get invoked by the operating system or application when a password change is being processed. Typically, the plug-in is used to enforce additional password rules. In the Identity Management space, these plug-ins are used to capture the password and push it to the Identity Server system.

Most targets reply on the status returned by the plug-in in order to complete the password operation. For example, on MS AD, the plug-in is invoked while a user is changing his/her password. The plug-in must return with a status in order for the user to see the response. In

addition, the plug-in response determines if AD will change the password of not. If the plug-in returns an error, the user will see an error message and the password change will fail. If the plug-in returns success, the password will be changed in AD. Other targets may behave differently.

Additional caution must be taken when a plug-in communicates with the IGI server. What happens if the plug-in fails to login to IGI? What happens if the IGI server is down? And so on. Reverse Password Sync plug-ins must provide configuration items to allow users to change passwords on the target in the event of communication errors with the IGI server. For example, the following list of configuration items should be considered.

- The plug-in fails to login to IGI
- IGI returns an error response for specific API
- The IGI response time is too long

By providing these configuration items, customers can choose the behavior when IGI is not reachable for any reasons.

Refer to Figure 2 – Reverse Password Sync Plug-in data flow for a description logic that the plug-in should follow. Additional error checking can be added based on the target. The figure references 4 APIs that are provided by IGI. These APIs are stuffiest to accomplish the reverse password sync plug-in, however, any published IGI REST API can be used to provide additional features. Note that response time is crucial, additional REST APIs will impact the end user experience on the target. The reverse password sync plug-in must be efficient and reliable.

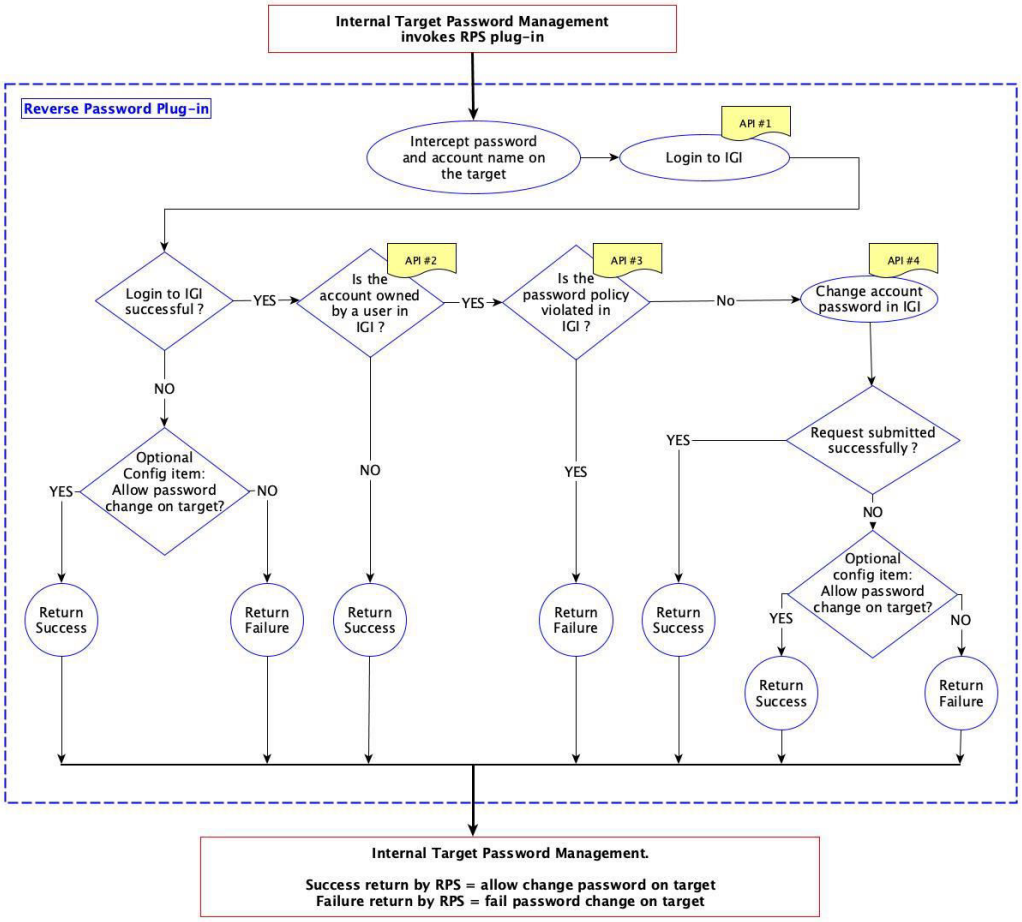


Figure 2 – Reverse Password Sync Plug-in data flow

Configuration Parameters

Each reverse password sync plug-in must have the following information in order to communicate with the IGI server and issue a password change. This information can be provided in many ways, such as configuration files, at installation time:

- IGI user login name
- IGI user password
- The URL of IGI REST server
- Name of the account configuration for the target where the reverse password sync plug-in is deployed on.
    - Remember that this target must be managed by IGI.

REST API Reference

This section describes the APIs that are referenced in Figure 2 – Reverse Password Sync Plug-in data flow.

API #1: Login

Login to IGI system and obtains token needed to call next methods

GET https://<VA_IP>:<VA_PORT>/igi/v2/security/login

Headers

Header	Description
realm	Realm
authorization	Basic Authentication

API #2: Find User Account

Finds one or more Account objects associated to a user through a SCIM search request.

POST https://<VA_IP>:<VA_PORT>/igi/v2/agc/users/accounts/.search

Headers

Header	Description
realm	Realm
authorization	Bearer Authorization
content-Type	application/scim+json

Request Body

Name	Type	Optional	Description
urn	String	No	urn:ietf:params:scim:api:messages:2.0:SearchRequest

```
*Note: find the Ideas account of the user with userName equals to \"jdoe\"
POST https://www.example.com:9343/igi/v2/agc/users/accounts/.search
POST_DATA
{
               "schemas": ["urn:ietf:params:scim:api:messages:2.0:SearchRequest"],
               "filter" : "urn:ibm:params:scim:schemas:resource:bean:agc:2.0:Account:person_code eq \"jdoe\" and
                        urn:ibm:params:scim:schemas:resource:bean:agc:2.0:Account:pwdcfg_name eq \"Ideas\""
      }
```

API #3: Validate Password

Checks if the password specified is compliant with the password policies of a specific account (see Get Password Policy REST API for password rules).

POST
https://<VA_IP>:<VA_PORT>/igi/v2/agc/users/accounts/{account_id}/password/check

Headers

Header	Description
realm	Realm
authorization	Bearer Authorization
content-Type	application/scim+json
cache-control	no-cache

Attributes

Attribute	Description
codeOperation	transaction key that identifies the operation and who performed it (optional parameter). For reverse password sync, if the value is specified please prefix the value with “TARGET”. For example, “TARGET_winAD_reverse_pwd_sync”

Parameters

Parameter	Description
account_id	Specifies the user account id

Request Body

Name	Type	Optional	Description
urn	String	No	urn:ibm:params:scim:api:messages:2.0:ChangePwd

API #4: Change Account Password

Changes the password of a specific account. The IGIPwd field represents the IGI current password for that account.

POST https://<VA_IP>:<VA_PORT>/igi/v2/agc/users/accounts/{account_id}/password

Headers

Header	Description
realm	Realm
authorization	Bearer Authorization
content-Type	application/scim+json
cache-control	no-cache

Attributes

Attribute	Description
codeOperation	transaction key that identifies the operation and who performed it (required parameter for reverse synch password). For reverse password sync, the value is specified and prefixed with “TARGET”. For example, “TARGET_winAD_reverse_pwd_sync”. If codeOperation with “TARGET” is not specified, another Change Password event will be created in the OUT Event queue in IGI for this account.

Parameters

Parameter	Description
account_id	Specifies the user account id

Request Body

Name	Type	Optional	Description
urn	String	No	urn:ibm:params:scim:api:messages:2.0:ChangePwd

```
POST https://www.example.com:9343/igi/v2/agc/users/accounts/663/password
POST_DATA:
        {
                "schemas" : ["urn:ibm:params:scim:api:messages:2.0:ChangePwd"],
                "IGIPwd":"IGI_current_password",
                "newPassword":"mynewpassword"
        }
```

Development Prerequisites

Creating a Reverse password Sync plug-in is a development process that requires knowledge in the following areas:

- Programming languages such as Java, C++, …
- Basic IGI administration
- RESTful APIs
- REST client development

- SSL Certificates
- In depth knowledge of password management on the target where the Reverse password Syncplug-in will be developed and deployed

ReversePwdSyncTutorial.zip package

The *ReversePwdSyncTutorial.zip* package contains the following files:

File name	Description
ApiManager.java	Java class that creates REST request and calls IGI REST APIs.
PwdSyncClient.java	The client java program that finds user's account, validates the password, and calls IGI to synchronize the password.
RequestComponents.java	Java class that holds information of the config.properties file.
config.properties	It contains IGI REST URL, login ID, password, and target name (accountConfig). See below for description of each property in the file.
pom.xml	An XML file that contains information about the project and configuration details used by Maven to build the project.

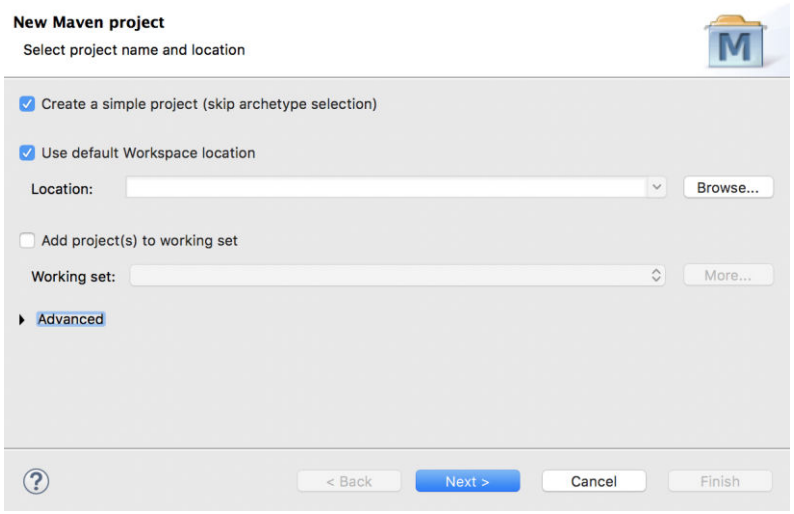
Table 2 - ReversePwdSyncTutorial.zip package

Sample Rest Client Prerequisites

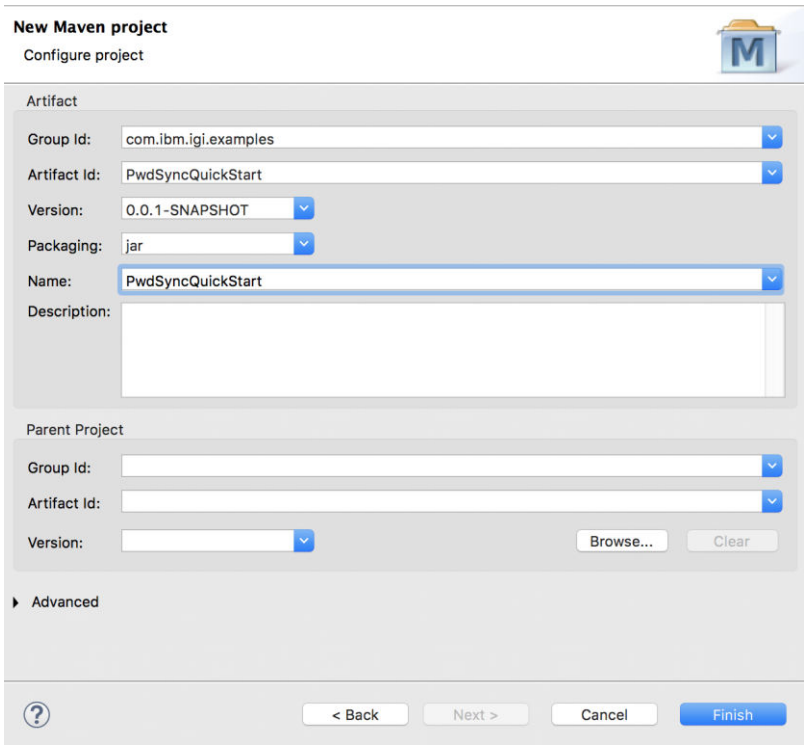
1. IBM JRE (SR3 FP10 or above)
2. Eclipse IDE for Java EE Developers.
3. Maven (a build automation tool used primarily for Java projects)

Steps for creating a sample Rest Client Application

1. Download Eclipse IDE for Java EE Developers.
2. Unzip the ReversePwdSyncTutorial.zip file.
3. In Eclipse, create a new Maven Project (File->New->Maven Project)
    a. Select "Create a simple project (skip archetype selection)" checkbox and click Next

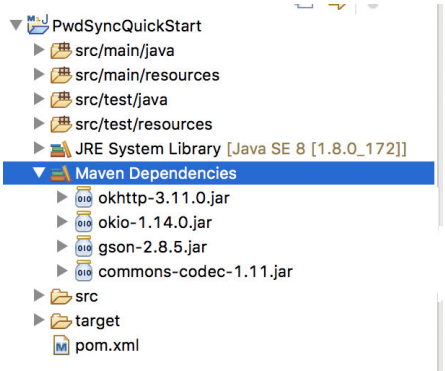


b. Specify “Group Id”, “Artifact Id”, and “Name” for your project. The artifact ID is your project name. You can enter any group ID. But preferably use the names used in this documentation. Click Finish after entering the project information.

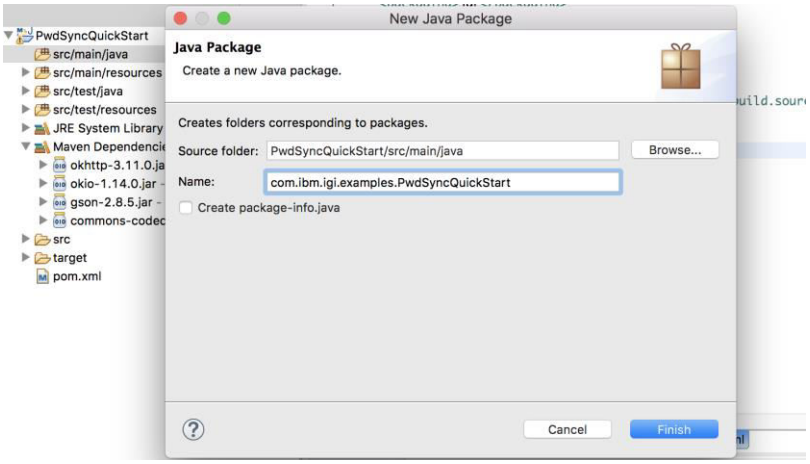


c. Configure the project build path to update the JRE System library being used. Make sure that you are using the latest one. By default, a 1.8 JRE library might appear in the build path. Remove that and Add the latest one, or use the workspace default (if that is Java 1.8 or later)

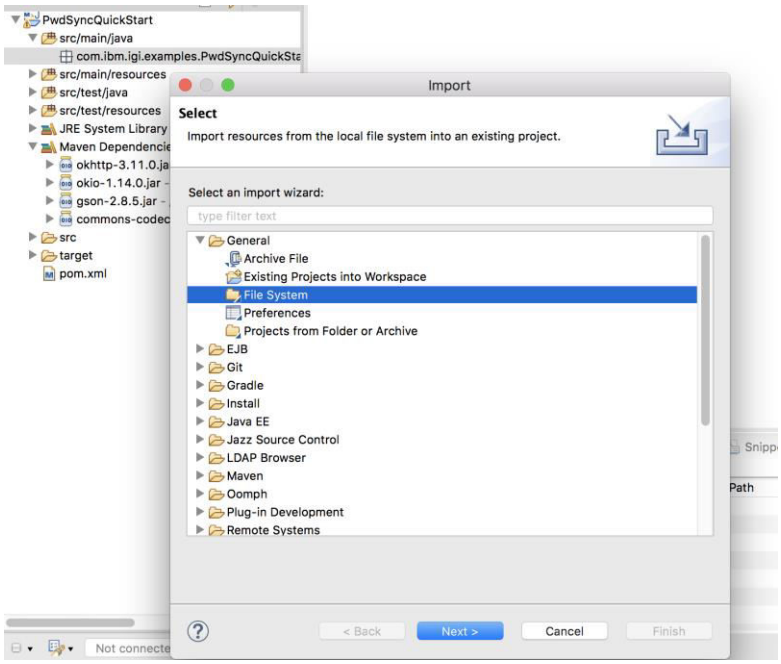
4. From the zip file, copy and paste the contents of the pom.xml file to the pom.xml file of your project. (Note: If you used different Group Id/Artifact Id than the above one, adjust those values accordingly in the pom.xml file)
5. Save the pom.xml file and let the project build. Observe under Maven Dependencies, the libraries being included after the build.

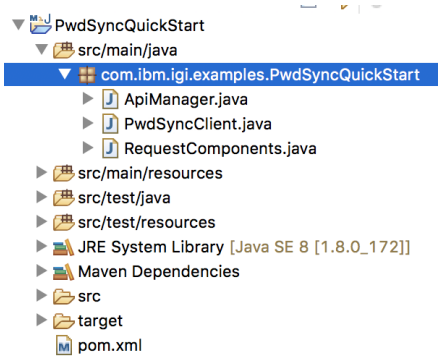


6. Select the src/main/java folder and create a package named “com.ibm.igi.examples.PwdSyncQuickStart”.

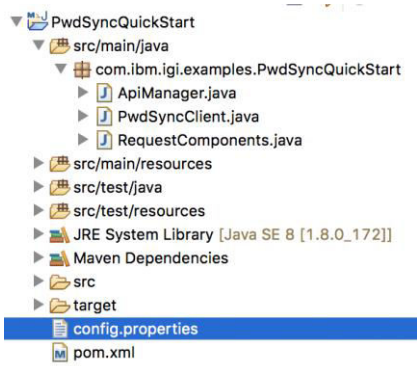


7. Import RequestComponents.java, ApiManager.java, and PwdSyncClient.java to "com.ibm.igi.examples.PwdSyncQuickStart" package.
    a) Right click on "com.ibm.igi.examples.PwdSyncQuickStart" package and select Import > General > File System > Next > Browse
    b) Select ReversePwdSyncTutorial folder extracted from the downloaded zip file
    c) Select the above three Java files and click Finish.





8. Under the project, import the config.properties file
   a) Right click on PwdSyncQuickStart project and select Import > General > File System > Next > Browse
   b) Select ReversePwdSyncTutorial folder extracted from the downloaded zip file
   c) Select config.properties file and click Finish.



9. Under the project, edit the config.properties file. Replace the following bold text to match your Identity Governance Intelligence environment.

```
igi.rest.api.login=admin
igi.rest.api.password=admin
igi.rest.api.url=https://9.1.1.1:9343/igi/v2
igi.pwdsync.accountConfig=pwd sync test
```

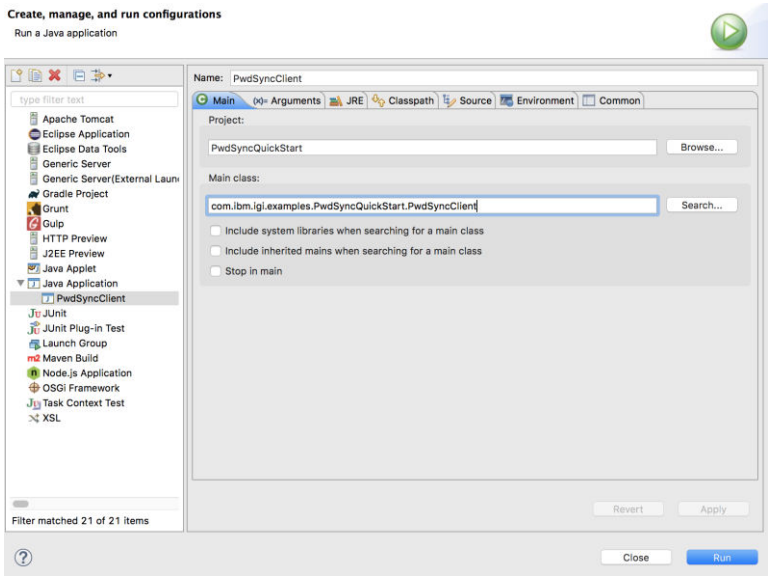
Description of properties of config.properties

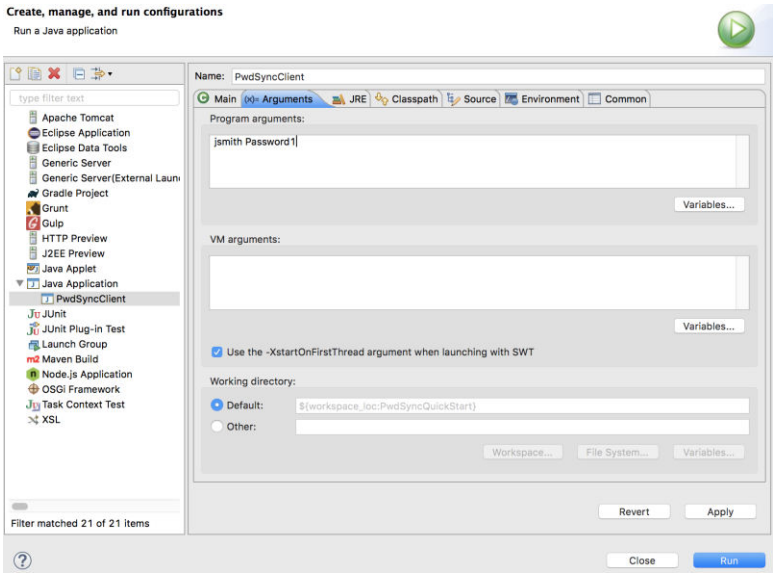
Property name	Description
igi.rest.api.login	The REST API login user. Example: admin
igi.rest.api.password	The password for the REST API login user. Example: admin
igi.rest.api.url	The URL of IGI REST server
igi.pwdsync.accountConfig	Name of the account configuration for the target that you added to the password sync group in the Setup step 3. See Table 1 - Password Synchronization Setup

10. Run the PwdSyncClient file as a Java Application

a) In Eclipse, go to Run->Run Configuration->Java Application->PwdSyncClient, and add the following to the program arguments:
<userID> <newPassword>

where the <userID> is the user ID of the account that you changed the password and the <newPassword> is the new password for the account.





Now you can log on IBM Security Identity Governance and Intelligence (IGI) and navigate to Access Governance Core > Monitor > “OUT events” to verify if there are “change password” events for user’s other accounts that belong to the same password synchronization group.