



Multi-Connector Adapter for IdentityIQ

PS Labs Connectivity

Table of Contents

1	Introduction	4
1.1	Terminology	4
1.2	Revision History	4
2	Connector Basics	6
2.1	Requirements	6
2.2	Inner Workings	6
2.2.1	Object Name Format	7
2.2.2	Partitioning	7
2.2.3	Aggregation Retry	8
2.2.4	Delta Aggregation	8
2.2.5	Threaded Provisioning	9
2.2.6	Threaded Connection Testing	9
2.2.7	Simulated Aggregation	9
2.3	Available Templates	10
2.3.1	Multi - Linux, Multi - AIX and Multi - Solaris	10
2.3.2	Multi - Windows Local - Direct	11
2.3.3	Multi-SQL Server Template	13
2.3.4	Multi-Sybase Database Template	13
2.3.5	Multi-Oracle Database Template	14
2.3.6	Multi-MySQL and MariaDB Database Templates	15
2.3.7	Multi-PostgreSQL Template	17
2.3.8	Multi-IBM DB2 Template	19
2.3.9	Creating a Custom Template	20
2.3.10	Make an Existing Application Multi-Capable	21
3	Configuration	22
3.1	General	22
3.2	Target-Specific Configuration	29

3.2.1	Embedded Configuration	29
3.2.2	Custom Object	30
3.2.3	Configuration Rule	31
3.2.4	Credential Rule	32
3.3	Provisioning	33
3.3.1	Before Provisioning Rule	33
3.3.2	Special account/group attributes	33
3.3.3	Account Provisioning Form	34
3.4	Credential Manager Integration	36
3.5	Auditing	37
4	Rule Examples	38
4.1	Multi Configuration Rule	38
4.2	Multi Credential Rule	39
4.3	Before Provisioning Rule	39
5	Support Plugin	41
5.1	Store Aggregation Results	41
5.2	Background Connection Test Task	41
5.2.1	Pre-Test Rule	48
5.2.2	Post-Test Rule	48
5.2.3	Connection Test Scenarios	49
5.3	Partitioned Group Aggregation	50
5.4	Auto-Correct LCM Provisioning Workflow Wrapper	52
5.4.1	Alternative Implementation Options	53
5.5	Auditing	54
5.5.1	Audit Instance Tag	54
5.6	Reporting	55
5.6.1	Multi-Connector Adapter Support Plugin Test Results Report	55

6	Plugin Request Executor	57
7	Known Limitations	58
7.1	Cloud Gateway	58
7.2	Delta Aggregation	58
8	Frequently Asked Questions	59
8.1	Which application types can be used with the Multi-Connector Adapter?	59
8.2	Does the adapter run on a separate host, or does it need to be installed on every IdentityIQ instance?	59
8.3	Are there any specific deployment considerations (high-availability, network latency, etc.)?	59
8.4	Are there data volume limits or performance considerations?	60
8.5	How to Avoid the “Fat Identity” Problem	61
8.6	Which logging mechanism does the Multi-Connector Adapter use?	61

1 Introduction

The connector that is described in this document is an adapter that can be used on top of another connector, allowing the configuration of a single application definition to manage multiple systems of the same type.

A common use case is where customers have hundreds or thousands of unix servers of the same vendor and need to manage these. Using a list of systems and their IP addresses and other system specific attributes, a single application definition in IdentityIQ can be used to manage access to all of these systems.

1.1 Terminology

A Multi-Connector application contains a generic definition and uses a list of attributes that define the systems or targets that it manages. The words 'system', 'target' and 'host' are often used to indicate the same things.

1.2 Revision History

Revision Date	Written/Edited By	Comments
Jan 2020	Menno Pieters	Initial release.
May 2020	Menno Pieters	Options and examples added.
October 2020	Menno Pieters	Additional templates.
March 2021	Menno Pieters	Frequently asked questions.
May 2021	Menno Pieters	Information about the object name format.
September 2021	Menno Pieters	Multi-threaded connection test, Multi-Connector Adapter Support Plugin, Auditing Features.
April 2022	Menno Pieters	Added MySQL and PostgreSQL. Additional features of the Multi-Connector Adapter Support Plugin. Pre- and Post-Test Rules, Pre- and Post-Test Rule Arguments.

May 2022	Menno Pieters	Added additional parameters for connection testing.
July 2022	Menno Pieters	Test Connection report in plugin.
August 2023	Menno Pieters	MySQL and MariaDB Update. Sybase added.

2 Connector Basics

2.1 Requirements

The connector only works correctly if the systems that it is supposed to manage are setup in the same way:

- The systems are of the same base type.
- The systems share the exact same schema.
- The provisioning policy for the systems is the same.
- Most of the other configuration settings are the same, except for connection information and credentials.

2.2 Inner Workings

The connector is an adapter. This means that it sits between IdentityIQ and the actual connector. This is comparable to connectors that are based on the OpenConnector framework¹, or to the LogiPlex connector in “adapter mode”. An adapter intercepts provisioning requests and modifies them before passing them on to the next connector class.

For aggregation, it receives information from the other connector class and modifies the information such that IdentityIQ understands it. This connector adapter will, for aggregation, first create a list of systems to connect to. Then for each target system, it will instantiate the “real” connector with the target-specific configuration. While reading, each account, group or entitlement will be changed such that identifying information (name, display name) will be unique across all targets, by adding a target identifier. How this is done, depends on the selected object name format.

When provisioning is requested, the target identifier is stripped from accounts, groups, and entitlements. Then, the connector is instantiated for each target in the plan. The plan is thus split – if necessary – and provisioned through multiple instances of the same connector, each with target-specific settings.

¹ IdentityIQ has connectors based on the SailPoint AbstractConnector framework and the OpenConnector framework. For the latter, there is a connector class that acts as an adapter to allow them to be used like the connectors based on the SailPoint AbstractConnector framework.

2.2.1 Object Name Format

On aggregation, the object name format is used to add the name of the system to the name of the object (account or group). Several formats are supported and are applied to both accounts and groups in the same way.

Format	Example	Description
AT	object@server	Similar to an email address. This is the default format.
BACKSLASH	server\object	Similar to Windows domain objects.
BLOCK	[server]object	The object name is prefixed with the server's name between square brackets.
COLON	server:object	The object name is prefixed with the server's name and a colon.
HASH	object#server	A hash character and the server's name are appended to the object name.
SLASH	server/object	Similar to the BACKSLASH format, but with a forward slash.
LDAP	object,target=server	(Since version 2022.06.30-0002) To support a more "natural" looking format for multiple directories.

It is important to choose the correct format, as during provisioning the name is split into the object and server name again. If the selected format could be ambiguous, the split may not happen correctly or not at all.

For example, if the username in a system is an email address, one should not select the @-format, as it would result in two @-signs: user@domain@server. A better option would perhaps be the block format ([server]user@domain) or hash format (user@domain#server).

2.2.2 Partitioning

If partitioning is enabled for the aggregation task, the list of servers will be divided over the suggested number of partitions, unless there are less servers than partitions. In that case each server will have its own partition.

Partitioning is not propagated to the target connectors. With partitioning, multiple systems can be aggregated in parallel.

2.2.3 Aggregation Retry

The connector creates a list of systems to aggregate from. This list is placed in a queue. If aggregation fails for a specific target, and retries are enabled, the system is added back to the end of the queue until it succeeds or the maximum number of retries is reached. The number of retries can be:

- 0: retries are disabled.
- 1 – 1000: retries are enabled up to the configured amount
- -1: retries are enabled with no limit (careful: dangerous option – aggregation may never end)

Every time a system is tried again, before starting a short pause is introduced. The pause is incremented in steps of 100 ms up to 300 times 100 ms (30000 ms = 30 seconds).

Whether or not aggregation is retried, depends on the error message and the errors messages defined as retryable in the application XML. Within the attributes map, an entry can be defined with substrings to match against the errors reported.

```
<entry key="multiRetryableAggregationErrors">
  <value>
    <List>
      <String>TimeoutException</String>
      <String>Connection refused</String>
    </List>
  </value>
</entry>
```

2.2.4 Delta Aggregation

The connector adapter does not support real delta aggregation, and delta aggregation is not propagated to the target connectors. However, delta aggregation can be used to only aggregate a specific set of target systems. To enable this, there are three options that can be used to include certain target systems:

- **Delta Aggregation Host Filter:** a list of target systems to include.
- **Delta Aggregation Include Previously Failed:** if enabled, the list of previously failed systems will be included, for a retry

- **Delta Aggregation Include Aggregated More Than ... Days:** set a non-zero number of days to include target systems that have not been aggregated for at least that amount of days.

The three options can be combined. If the filter is empty and the other two are not selected, the aggregation will aggregate from all systems, but deletions will not be processed.

This delta aggregation can be combined with partitioning.

2.2.5 Threaded Provisioning

Provisioning can be executed in multiple threads. A provisioning plan can contain account or group provisioning requests for multiple target systems. If that is the case, the list can be divided and handled by a configurable number of threads in parallel. The maximum allowed number of threads is 16, the default is 1.

2.2.6 Threaded Connection Testing

Since version 2021.08.27-0001, the connection test can be performed in a multi-threaded fashion, allowing several of target systems to be tested in parallel. If the number of threads is set to 1 (default), multi-threading will not be used. The first system to fail will cause connection testing to abort. The maximum number of threads is 64. Typically, no more than twice the number of CPU cores should be configured.

2.2.7 Simulated Aggregation

Since version 2022.08.30-0004, there is an option to simulate aggregation for failed targets. This 'simulation' works together with the retry mechanism. If, after the configured number of retries (minimum 1), no successful aggregation can be performed from the target system, the connector will simply return what it knows, by listing accounts or groups stored in the IdentityIQ database.

By doing this, the aggregation process 'thinks' that aggregation was successful, allowing the processing of deletions from target systems that have actually been aggregated successfully.

If audit records are to be created for failed targets on aggregation, these audit records will still be created, so failed aggregations can be tracked. The aggregation task, however, will report successful completion.

2.3 Available Templates

Several templates are available, including *Multi – Linux* and *Multi – Windows Local – Direct*, *Multi – Oracle* and *Multi – SQL Server*. This list is growing over time, and it is relatively easy to create other templates.

2.3.1 Multi – Linux, Multi – AIX and Multi – Solaris

These application templates are configured for Linux, AIX and Solaris servers. Most information is pre-configured, but for each managed system, a few attributes must be provided:

- Hostname or IP address
- Port (default 22)
- Login information: username, password or SSH key file path and SSH key password
- Whether the user is a sudo user or not (root).

Attribute	Type	Description
<code>IsSudoUser</code>	boolean	Tells whether the user must use the <code>sudo</code> command to perform privileged commands
<code>PassphraseForPrivateKey</code>	password	If an SSH key is used for login, this attribute contains the encrypted password.
<code>PrivateKeyFilePath</code>	string	If an SSH key is used for login, this attribute contains the path to the file.
<code>SshPort</code>	integer	The port number for the SSH connection. The default is 22 but can be overridden using this attribute.
<code>SudoUser</code>	string	The name of the service account to login to the Linux server and if necessary, to use the <code>sudo</code> command.
<code>SudoUserPassword</code>	password	The encrypted password if password authentication is used.

Attribute	Type	Description
host	string	The hostname or IP address of the managed system.

For requirements, features, etc. refer to the connector guides for the respective connectors:

- IBM AIX:
<https://documentation.sailpoint.com/connectors/identityiq/ibm/aix/help>
- Linux – Direct:
<https://documentation.sailpoint.com/connectors/identityiq/linux/help>
- Solaris – Direct:
<https://documentation.sailpoint.com/connectors/identityiq/oracle/solaris/help>

2.3.2 Multi – Windows Local – Direct

The application template is configured for Windows servers. Most information is pre-configured, but for each managed system, a few attributes must be provided:

- Hostname or IP address of the IQService
- IQService Port (default 5050)
- (Optional) IQService User and Password
- Whether or not to use TLS for the connection to IQService
- Login information: username, password
- Windows server hostname

Attribute	Type	Description
IQServiceHost	string	The host name or IP address of the server running IQService.
IQServicePort	integer	The TCP port number for the IQService service.
IQServiceUser	string	The username for the IQService, if authentication is configured.

Attribute	Type	Description
IQServicePassword	password	The password for the IQService user. The password can or should be encrypted.
useTLSForIQService	boolean	A flag to indicate whether or not to use TLS encrypted connections to the IQService. This requires additional configuration of the IQService to be enabled.
user	string	The windows or active directory username of the service account to login to the Windows server.
password	password	The encrypted password if password authentication is used.
server	string	The hostname or IP address of the managed system.
pageSize	integer	Number of objects to fetch in a single request. Defaults to 1000.
disableQualifyingLocalObjects	boolean	Flag to indicate whether aggregated objects must not be prefixed with server name. (Defaults to false. If set to true then aggregated object will not be prefixed with server name). Example: False: WinServer\JohnDoe True: JohnDoe

For requirements, features, etc. refer to the **Windows Local – Direct** Connector Guide: https://documentation.sailpoint.com/connectors/identityiq/microsoft/windows_local/help/integrating_windows_local/intro.html.

2.3.3 Multi-SQL Server Template

The template “Multi – Microsoft SQL Server – Direct” can be used to configure multiple SQL Server instances. Most information is pre-configured, but for each managed system, a few attributes must be provided:

- JDBC URL for the SQL Server instance
- Username
- Password
- (Optional) Driver class, but could also be provided globally

Attribute	Type	Description
url	string	The JDBC URL for the managed system.
username	string	The local or AD username for the database. This must be a user with administrative access.
password	string	The password for the user. This can be encrypted.
driverClass	string	The class for the JDBC driver, typically <code>com.microsoft.sqlserver.jdbc.SQLServerDriver</code> .

For requirements, features, etc. refer to the **Microsoft SQL Server – Direct** Connector Guide:

https://documentation.sailpoint.com/connectors/identityiq/microsoft/sql_server/help/integrating_ms_sql_server/introduction.html.

2.3.4 Multi-Sybase Database Template

The template “Multi – Sybase – Direct” can be used to configure multiple SQL Server instances. Most information is pre-configured, but for each managed system, a few attributes must be provided:

- JDBC URL for the SQL Server instance.
- Username.
- Password.
- (Optional) Driver class but could also be provided globally.

Attribute	Type	Description
url	string	The JDBC URL for the managed system.

Attribute	Type	Description
user	string	The local or AD username for the database. This must be a user with administrative access.
password	string	The password for the user. This can be encrypted.
driverClass	string	The class for the JDBC driver, typically <code>com.sybase.jdbc2.jdbc.SybDataSource</code> .
includeDatabases	string	Names of databases to include (mutually exclusive with <code>excludeDatabases</code>).
excludeDatabases	string	Names of databases to exclude (mutually exclusive with <code>includeDatabases</code>).

For requirements, features, etc. refer to the **SAP Sybase ASE (Sybase – Direct)** Connector Guide:

https://documentation.sailpoint.com/connectors/identityiq/sap/sybase/help/integrating_sap_sybase_sap_ase/intro.html

2.3.5 Multi-Oracle Database Template

The template “Multi – Oracle Database – Direct” can be used to configure multiple Oracle database server instances. Most information is pre-configured, but for each managed system, a few attributes must be provided:

- JDBC URL for the Oracle Server instance.
- Username.
- Password.
- (Optional) Driver class but could also be provided globally.
- (Optional) Additional driver parameters.

Attribute	Type	Description
url	string	The JDBC URL for the managed system.
user	string	The administrative username for the database.

Attribute	Type	Description
password	string	The password for the user. This can be encrypted.
driverClass	string	The class for the JDBC driver, typically <code>oracle.jdbc.driver.OracleDriver</code> .
additionalConnProperties	string	Additional JDBC driver parameters.

For requirements, features, etc. refer to the **Oracle Database – Direct** Connector Guide:

https://documentation.sailpoint.com/connectors/identityiq/oracle/database/help/integrating_oracle_database/introduction.html.

2.3.6 Multi-MySQL and MariaDB Database Templates

The Multi-MySQL and Multi-MariaDB Database Templates are based on the Professional Services Extensions **MySQL – Direct Connector**, **MySQL 5 – Direct Connector**, **MariaDB 10 – Direct Connector** and **MariaDB 11 – Direct Connector**. These are required for the multi-variants of these connectors to be installed and used.

Most information is pre-configured, but for each managed system, a few attributes must be provided:

- For each schema type:
 - JDBC URL for the MySQL Server instance
 - Username
 - Password
 - (Optional) Driver class, but could also be provided globally

Attribute	Type	Description
url	string	The JDBC URL for the managed system.
user	string	The administrative username for the database.
password	string	The password for the user. This can be encrypted.

Attribute	Type	Description
<code>driverClass</code>	string	The class for the JDBC driver, typically <code>com.mysql.cj.jdbc.Driver</code> . For older JDBC drivers, this may be <code>com.mysql.jdbc.Driver</code> .
<code>database_access.url</code>	string	The same as <code>url</code> above.
<code>database_access.user</code>	string	The same as <code>user</code> above.
<code>database_access.password</code>	string	The same as <code>password</code> above.
<code>database_access.driverClass</code>	string	The same as <code>driverClass</code> above.
<code>role_access.url</code>	string	The same as <code>url</code> above*.
<code>role_access.user</code>	string	The same as <code>user</code> above*.
<code>role_access.password</code>	string	The same as <code>password</code> above*.
<code>role_access.driverClass</code>	string	The same as <code>driverClass</code> above*.
<code>table_access.url</code>	string	The same as <code>url</code> above.
<code>table_access.user</code>	string	The same as <code>user</code> above.
<code>table_access.password</code>	string	The same as <code>password</code> above.
<code>table_access.driverClass</code>	string	The same as <code>driverClass</code> above.
<code>user_access.url</code>	string	The same as <code>url</code> above.
<code>user_access.user</code>	string	The same as <code>user</code> above.
<code>user_access.password</code>	string	The same as <code>password</code> above.
<code>user_access.driverClass</code>	string	The same as <code>driverClass</code> above.

* only applies to MySQL 8 (not yet implemented) and MariaDB 11.

The connector is built using the out-of-the-box **JDBC** Connector. See the **JDBC** Connector Guide for more information:

https://documentation.sailpoint.com/connectors/identityiq/jdbc/help/integrating_jdbc/introduction.html.

2.3.6.1 Service Account Privileges

The following describes the necessary access rights for the service account for IdentityIQ on each of the managed systems.

2.3.6.1.1 Read-Only Scenario (Aggregation)

When IdentityIQ will only aggregate, it will need read permissions on the database. Assuming that the service account is `identityiq@%`, the following command can be used to grant the necessary access as the root (or similar) user:

```
GRANT SELECT ON *.* TO 'identityiq'@'%';
```

2.3.6.1.2 Read-Write Scenario (Provisioning)

When IdentityIQ also needs to provision, the service account needs full administrative access. The reason is that one cannot grant access to a user that one does not have oneself. In this case, it does make sense to restrict the access to specific IP addresses. This is done by using the IP address, hostname or IP range as part of the username: `identityiq@10.1.1.2.3`.

To grant full access as the root user (or similar), the following command can be used:

```
GRANT ALL PRIVILEGES ON *.* TO 'identityiq'@'10.1.1.2.3';  
GRANT GRANT OPTION ON *.* TO 'identityiq'@'10.1.1.2.3';
```

2.3.7 Multi-PostgreSQL Template

The Multi-PostgreSQL Database Template is based on the Professional Services Extension **PostgreSQL - Direct Connector** and requires this connector to be installed before the Multi-PostgreSQL Database can be used.

Most information is pre-configured, but for each managed system, a few attributes must be provided:

- For each schema type:
 - JDBC URL for the MySQL Server instance
 - Username
 - Password
 - (Optional) Driver class, but could also be provided globally

Attribute	Type	Description
url	string	The JDBC URL for the managed system.

Attribute	Type	Description
user	string	The administrative username for the database.
password	string	The password for the user. This can be encrypted.
driverClass	string	The class for the JDBC driver, typically <code>org.postgresql.Driver</code> .
group.url	string	The same as <code>url</code> above.
group.user	string	The same as <code>user</code> above.
group.password	string	The same as <code>password</code> above.
group.driverClass	string	The same as <code>driverClass</code> above.
privileges.url	string	The same as <code>url</code> above.
privileges.user	string	The same as <code>user</code> above.
privileges.password	string	The same as <code>password</code> above.
privileges.driverClass	string	The same as <code>driverClass</code> above.
schema_access.url	string	The same as <code>url</code> above.
schema_access.user	string	The same as <code>user</code> above.
schema_access.password	string	The same as <code>password</code> above.
schema_access.driverClass	string	The same as <code>driverClass</code> above.
table_access.url	string	The same as <code>url</code> above.
table_access.user	string	The same as <code>user</code> above.
table_access.password	string	The same as <code>password</code> above.
table_access.driverClass	string	The same as <code>driverClass</code> above.

The connector is built using the out-of-the-box **JDBC** Connector. See the **JDBC** Connector Guide for more information:

https://documentation.sailpoint.com/connectors/identityiq/jdbc/help/integrating_jdbc/introduction.html.

2.3.7.1 Service Account Privileges

The following describes the necessary access rights for the service account for IdentityIQ on each of the managed systems.

2.3.7.1.1 Read-Only Scenario (Aggregation)

For read-only access, no special permissions are needed. A valid login account can access the necessary tables and views.

2.3.7.1.2 Read-Write Scenario (Provisioning)

For read-write the service account must be the equivalent of a 'superuser' in order to create other users and grant access to all database schemas.

2.3.8 Multi-IBM DB2 Template

The Multi-IBM DB2 template is based on the out-of-the-box IBM DB2 connector. There are only a few attributes that can be set per target:

- JDBC URL for the DB2 Server instance
- Username
- Password
- (Optional) Driver class, but could also be provided globally

Attribute	Type	Description
url	string	The JDBC URL for the managed system.
user	string	The administrative username for the database.
password	string	The password for the user. This can be encrypted.
driverClass	string	The class for the JDBC driver, typically <code>com.ibm.db2.jcc.DB2Driver</code> .

For requirements, features, etc. refer to the **IBM DB2** Connector Guide:

https://documentation.sailpoint.com/connectors/identityiq/ibm/db2/help/integrating_ibm_db2/intro.html.

2.3.9 Creating a Custom Template

All available application templates are stored in a Configuration object, named "ConnectorRegistry". New application types can be added to IdentityIQ by creating a new entry with an application template in this object. This requires a merge object to be defined in XML, in which the Application object is included.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE sailpoint PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<sailpoint>
  <ImportAction name="merge">
    <Configuration name="ConnectorRegistry">
      <Attributes>
        <Map>
          <entry key="applicationTemplates">
            <value>
              <List>
<!-- Your template goes here -->
              </List>
            </value>
          </entry>
        </Map>
      </Attributes>
    </Configuration>
  </ImportAction>
</sailpoint>
```

An existing application template can be copied into the XML object above, after which some changes need to be applied.

For example, the entry for Delimited Files is this:

```
<Application connector="sailpoint.connector.DelimitedFileConnector"
featuresString="DIRECT_PERMISSIONS, NO_RANDOM_ACCESS, DISCOVER_SCHEMA"
icon="enterpriseIcon" name="DelimitedFile Template" type="DelimitedFile">
  <Attributes>
    <Map>
      <entry key="formPath" value="delimitedAttributesForm.xhtml"/>
      <entry key="formPathRules" value="delimitedRulesForm.xhtml"/>
      <entry key="sftpAuthMethod" value="password"/>
    </Map>
  </Attributes>
</Application>
```

To make this work with the Multi-Connector-Adapter, a copy of the above needs to be injected into the merge-object and edited:

- The name must be changed
- The connector class must be changed
- The formPath and formPathRules need to be changed
- Additional attributes for the Multi-Connector Adapter need to be added with default values and one to specify the real, original connector to use.

- Possibly the `featureString` needs to be adapted: `DISCOVER_SCHEMA` may not work.
- Check that no credentials are defined in the template.

```
<Application connector="sailpoint.pse.connector.MultiConnectorAdapter"
featuresString="DIRECT_PERMISSIONS, NO_RANDOM_ACCESS" icon="enterpriseIcon"
name="Multi-DelimitedFile Template" type="Multi-DelimitedFile">
  <Attributes>
    <Map>
      <entry key="formPath" value="multiAppAttributes.xhtml"/>
      <entry key="formPathRules" value="multiAppRulesForm.xhtml"/>
      <entry key="sftpAuthMethod" value="password"/>
      <entry key="multiApplicationRealConnector"
value="sailpoint.connector.DelimitedFileConnector" />
      <entry key="multiRetryableAggregationErrors">
        <value>
          <List>
            <String>TimeoutException</String>
            <String>Connection refused</String>
          </List>
        </value>
      </entry>
    </Map>
  </Attributes>
</Application>
```

Other attributes listed in 3.1 can be added to the template with default values.

For the `formPathRules`, depending on the type of application, it may be necessary to create a variant of the `xhtml` file that combines the rule settings for both the “normal” application and the Multi-Connector Adapter.

2.3.10 Make an Existing Application Multi-Capable

It is also possible to directly make an existing application multi-capable. If you don’t need the template, you can define the default settings for an application and then make the changes to parameters and attributes as described in the previous subsection (2.3.9).

3 Configuration

3.1 General

The table below lists the configuration options. Some of these may not have a UI equivalent but need to be configured in the XML definition.

Configuration → Settings

UI Option	Attribute	Type	Description
-	<code>multiApplicationRealConnector</code>	String	The name of the connector to use for the target systems. This is the preferred way of specifying the connector. If not specified, it will try the template.
-	<code>multiApplicationTemplate</code>	String	(deprecated) This must point to a template from the ConnectorRegistry from which the connector and possibly other information will be retrieved.
Custom Object Name	<code>multiConfigurationCustom</code>	String	The name of a custom object that contains target server configurations.
Object Name Format	<code>multiObjectNameFormat</code>	String	The format option for combining the account/group name and server name. Six predefined formats are available. The default format is <code>name@server</code> . See 2.2.1.

UI Option	Attribute	Type	Description
Manage Display Name Format	<code>multiChangeDisplayName</code>	Boolean	This option determines whether or not the display name for accounts and groups will be rewritten during aggregation and provisioning. Default is true.
Maximum Test Connection Targets	<code>multiMaxTestConnections</code>	Integer	During connection test, the connector will contact target systems. To prevent this from taking too long, a limit is configured for the maximum number of targets to try. The default is set to 50.
Test Connection Threads	<code>multiTestConnectionThreads</code>	Integer	The number of threads to use for connection testing. If set to 2 or more, multiple threads will be used. All targets (filters applied), up to the configured maximum number of targets, will be tested and all errors are reported at the end.
-	<code>multiTestConnectionDetailedExceptions</code>	Boolean	If set to true, a full stack trace is printed for multi-threaded test connection failures, otherwise only a list of failed targets.

UI Option	Attribute	Type	Description
Test Connection Host Filter	multiTestConnectionFilter	String	<p>This field is expected to be a comma separated list of target names.</p> <p>If set, during test connection, only the systems listed will be evaluated, up to the limit for the Maximum Test Connection Targets. Targets that do not exist will be ignored.</p> <p>Note: the values are case-insensitive.</p>
Maximum Aggregation Retries	multiMaxAggregationRetries	Integer	<p>During aggregation, if a target fails, the aggregation can be retried until it succeeds.</p>
Delay Exceptions	multiDelayAggregationExceptions	Boolean	<p>For targets that have failed and cannot be retried, the exception can either be thrown immediately, causing aggregation (or the partition) to be aborted, or when this option is enabled, saved until the aggregation for all target completes. Then the exceptions are thrown at the end.</p> <p>This option only works if the value for 'Maximum Aggregation Retries' is set to at least 1.</p>

UI Option	Attribute	Type	Description
Simulate Aggregation for Failed Endpoints	multiSimulateAggregationOnFailure	Boolean	<p>For targets that have failed and cannot be retried anymore, the multi-connector adapter will look up accounts or groups/entitlements for the failed target and return these instead of information from the real target (see 2.2.7).</p> <p>This option only works if the value for '<i>Maximum Aggregation Retries</i>' is set to at least 1.</p>
Delta Aggregation Host Filter	multiDeltaAggregationFilter	String	<p>This field is expected to be a comma separated list of system names.</p> <p>If set, this list will be used during delta provisioning to filter which systems will be included. Systems that do not exist will be ignored.</p> <p>Note: the values are case-insensitive.</p>
-	multiRegisterAggregationResult	Boolean	<p>If set to true (default), after aggregation, the results will be stored on the application definition: a list of systems for which aggregation failed, and a timestamp for last successful aggregation for those which succeeded.</p>

UI Option	Attribute	Type	Description
Delta Aggregation Include Previously Failed	multiDeltaAggregationIncludeFailed	Boolean	If the options <code>multiRegisterAggregationResult</code> is enabled, the failed aggregations will be tracked in the application. With this option set to true, the failed systems will be retried as part of delta aggregation
Delta Aggregation Include Aggregated More Than ... Days	multiDeltaAggregationIncludeAged	Integer	If the options <code>multiRegisterAggregationResult</code> is enabled, a last successful aggregation date is kept in the application for each system. With this option set to a number of 1 or up, systems that have not been aggregated successfully in that number of days, will be included in the delta aggregation.
Provisioning Threads	multiMaxProvisioningThreads	Integer	Provisioning can be enabled to perform parts of the provisioning plan in parallel. With a value of 1 or less only a single thread is used, if more than 1, up to 16, at most the number of configured threads will be run in parallel.

UI Option	Attribute	Type	Description
-	multiRetryableAggregationErrors	List	In the XML, a list of retryable errors can be provided for aggregation. The list will be considered as case-insensitive substrings for the exception messages evaluated. See 2.2.3 .
-	auditTestConnectionFailure	Boolean	If set to 'true', audit records will be created with information about failed connection tests. This also requires the audit event Multi-Connector Adapter Failed Connection Test to be enabled (see 3.5)
-	multiAuditAggregationFailure	Boolean	If set to 'true', audit records will be created with information about failed connection tests. This also requires the audit event Multi-Connector Adapter Failed Aggregation to be enabled (see 3.5)

UI Option	Attribute	Type	Description
-	<code>multiThrowEmptyServerListException</code>	Boolean	<p>If no targets are defined for aggregation or connection testing, the default behavior since version 2022.06.23-0001 is to throw an exception. If this attribute is explicitly set to <code>false</code>, it will only log a warning message. Test connection and aggregation will end successfully. If not defined, the default is <code>true</code>.</p> <p>NOTE: if set to <code>false</code>, and if for any no targets are found in the embedded configuration, the custom object and/or configuration rule results, using the option 'Detect deleted accounts' or 'Detect deleted groups' will delete all accounts and groups during aggregation from IdentityIQ's database.</p>

Rules → MultiConnector Rules

UI Option	Attribute	Type	Description
Multi Configuration Rule	<code>multiConfigurationRule</code>	String	The name of the rule used to collect target configurations.
Multi Credential Rule	<code>multiCredentialRule</code>	String	The name of the rule used to look up and fill in

			credentials for the target configurations.
--	--	--	--

3.2 Target-Specific Configuration

The configuration for the targets to be handled by the adapter can be configured in three different ways:

- Statically embedded in the application definition,
- Statically embedded in a Custom object,
- Dynamically generated by a rule.

Any combination of the above is possible as well.

3.2.1 Embedded Configuration

A set of systems can be embedded in the application definition, within the Attributes map in an entry with the key “multiServerConfigurationMap”. The entry is expected to contain a list of map entries with server information.

Every map must contain the server specific connection information and the required attribute multiHostIdentifier with the target name. The target name must be a unique identifier but does not have to be the actual system name or IP address. That information belongs in the map.

An example:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Application PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Application connector="sailpoint.pse.connector.MultiConnectorAdapter"
featuresString="ENABLE, DISCOVER_SCHEMA, PROVISIONING, ADDITIONAL_ACCOUNT_REQUEST,
ACCOUNT_ONLY_REQUEST" icon="enterpriseIcon" name="MultiLinux" profileClass=""
type="MultiConnector">
  <Attributes>
    <Map>
...
      <entry key="multiServerConfigurationMap">
        <value>
          <List>
            <Map>
              <entry key="IsSudoUser">
                <value>
                  <Boolean>true</Boolean>
                </value>
              </entry>
              <entry key="PassphraseForPrivateKey" value="*****"/>
              <entry key="PrivateKeyFilePath"/>
              <entry key="SshPort" value="22"/>
              <entry key="SudoUser" value="sailpoint"/>
            </Map>
          </List>
        </value>
      </entry>
    </Map>
  </Attributes>
</Application>
```

```

        <entry key="SudoUserPassword" value="*****"/>
        <entry key="host" value="172.16.8.129"/>
        <entry key="multiHostIdentifier" value="LinuxMini1"/>
    </Map>
</List>
</value>
</entry>

```

...

3.2.2 Custom Object

A custom object can be used to configure the systems to handle. The custom object will contain an entry per system. Each entry is expected to be a map containing system specific settings to override the defaults, primarily connection information.

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Custom PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Custom name="MultiConfig-MultiLinux">
  <Attributes>
    <Map>
      <entry key="LinuxMini1">
        <value>
          <Map>
            <entry key="IsSudoUser">
              <value>
                <Boolean>true</Boolean>
              </value>
            </entry>
            <entry key="PassphraseForPrivateKey" value="*****"/>
            <entry key="PrivateKeyFilePath"/>
            <entry key="SshPort" value="22"/>
            <entry key="SudoUser" value="sailpoint"/>
            <entry key="SudoUserPassword" value="*****"/>
            <entry key="host" value="172.16.8.129"/>
          </Map>
        </value>
      </entry>
      <entry key="LinuxMini2">
        <value>
          <Map>
            <entry key="IsSudoUser">
              <value>
                <Boolean>true</Boolean>
              </value>
            </entry>
            <entry key="PassphraseForPrivateKey" value="*****"/>
            <entry key="PrivateKeyFilePath"/>
            <entry key="SshPort" value="22"/>
            <entry key="SudoUser" value="sailpoint"/>
            <entry key="SudoUserPassword" value="*****"/>
            <entry key="host" value="172.16.8.130"/>
          </Map>
        </value>
      </entry>
    </Map>
  </Attributes>
</Custom>

```

3.2.3 Configuration Rule

A rule can be used to dynamically generate a list, for example by reading information from an external system. The rule is expected to return a list of maps, similar to the information embedded in the application definition.

An extremely simple example would be as follows:

```
List results = new ArrayList();
Map system = new HashMap();
system.put("IsSudoUser", true);
system.put("PassphraseForPrivateKey", "*****");
system.put("SshPort", "22");
system.put("SudoUser", "sailpoint");
system.put("SudoUserPassword", "*****");
system.put("host", "172.16.8.129");
system.put("multiHostIdentifier", "LinuxMinil");
results.add(system);
return results;
```

A more advanced example is provided in 4.1 .

An important note on this rule is that any exception happening during the rule execution should be passed back to the connector. If errors are caught, e.g. the source systems stops responding, and an incomplete list of configurations is returned, this could result in loss of account and/or group information in IdentityIQ.

The rule will get, apart from the default `log` and `context` variable, the following inputs:

Variable	Type	Description
application	sailpoint.object.Application	The application definition of the base Multi-Connector Application.

Variable	Type	Description
<code>purpose</code>	String	<p>(version 2021.09.17-0001 and up)</p> <p>A string indicating why the list of targets is being requested. This can typically have the following values:</p> <ul style="list-style-type: none"> • <code>TEST_CONNECTION</code>: connection testing; • <code>AGGREGATE_ACCOUNT</code>: account aggregation (full or delta); • <code>AGGREGATE_GROUP</code>: group aggregation of any group type; • <code>GETOBJECT_ACCOUNT</code>: get a single account; • <code>GETOBJECT_GROUP</code>: get a single group object of any group type; • <code>PROVISION</code>: provisioning of any object type. <p>Another possible value would be <code>UNKNOWN</code>, but only in exceptional cases. When using the Multi-Connector Adapter Support Plugin's connection test task (5.2), a custom value can be entered.</p>

3.2.4 Credential Rule

For configurations stored in IdentityIQ or retrieved from a rule, but without credentials, it is possible to retrieve the credentials on the fly from an external system, using a rule. This is useful if credentials change periodically and need to be retrieved from for example a PAM solution.

The rule will get, apart from the default `log` and `context` variable, three more inputs:

Variable	Type	Description
<code>application</code>	<code>sailpoint.object.Application</code>	The application definition of the base Multi-Connector Application.
<code>server</code>	Map	A map with attribute names and values, specific for the managed system.
<code>serverName</code>	String	The name (identifier) for the managed system.

The expected output is a Map object with names of credential attributes and their values. The exact contents will differ for each system type.

An example is provided in 4.2 .

3.3 Provisioning

3.3.1 Before Provisioning Rule

A before provisioning rule can be used to make last minute changes to the provisioning plan, before it is handed to the adapter and connector. This can be useful to automatically correct a plan. It is possible that a user requests an entitlement on Server B (`group1@serverB`), but during the request, this is initially tied to the user's account on Server A (`user1@serverA`). The before provisioning rule can make use of code built into the connector to check and correct this.

See 4.3 .

3.3.2 Special account/group attributes

There are special attributes that can be used in the provisioning form and used to compose the username and possibly other purposes. These start with `IIQ_multi` and will be ignored (removed) by the connector before the plan is sent to the real connector.

So, any attribute name in the provisioning plan that starts with `IIQ_multi` is stripped from the plan and is only used for adapter-internal purposes.

3.3.3 Account Provisioning Form

Especially in cases of manual account creation, the provisioning form could be set up to allow the requester or approver or application owner to enter a plain username and select a target. For this, specialized fields can be added to the account provisioning form:

Attribute	Display Name	Type	Hidden	Description
IIQ_multiUserName	User Name	String	no	The plain username for the account on the target system.
IIQ_multiServerName	Server Name	String	no	The target system name for the account.
<i>(the attribute used as native identity)</i>	Composed User Name	String	yes	The username consisting of the local part and server name in the configured format, like <code>server\user</code> or <code>user@server</code> .

The fields `IIQ_multiUserName` and `IIQ_multiServerName` will be visible and should be marked as required, refresh the form on change and display only. Attributes starting with `IIQ_multi` will be ignored by the connector, even if not marked as display only.

Type Settings

☐ Multi-Valued
☒ Refresh on Change
☐ Authoritative

☒ Required
☐ Review Required
☒ Display Only

The field Composed UserName should be hidden or at least marked read-only so it cannot be changed, but will be built from the first two field once both have been filled in.

3.3.3.1 Allowed Values for Server Names

For the field `IIQ_multiServerName`, an Allowed Values Rule should be provided. A generic example of such a rule is provided here:

```
import sailpoint.connector.Connector;
import sailpoint.connector.ConnectorFactory;
import sailpoint.object.Application;
import sailpoint.tools.Util;
import sailpoint.web.group.AccountGroupDTO;

import sailpoint.pse.connector.MultiConnectorAdapter;

Application application = null;

if (group != void && group instanceof AccountGroupDTO) {
    AccountGroupDTO agd = (AccountGroupDTO) group;
    String applicationName = agd.getApplicationName();
    application = context.getObjectByName(Application.class, applicationName);
}

if (application == null && Util.isNotNullOrEmpty(field.getApplication())) {
    String applicationName = field.getApplication();
    application = context.getObjectByName(Application.class, applicationName);
}

if (application != null) {
    MultiConnectorAdapter connector = (MultiConnectorAdapter)
ConnectorFactory.createConnector("sailpoint.pse.connector.MultiConnectorAdapter",
application, null);
    List serverData = connector.getServerList();
    List serverNames = new ArrayList();
    for (Map data: serverData) {
        String name = data.get(MultiConnectorAdapter.ATTR_MULTI_HOSTID);
        if (Util.isNotNullOrEmpty(name)) {
            serverNames.add(name);
        }
    }
    Collections.sort(serverNames);
    return serverNames;
}
```

The example above uses the application definition to create a Multi-Connector Adapter and use a method on the adapter to get the configured server list. From that list, the identifiers are retrieved and listed.

3.3.3.2 Value Rule for Composed User Name

Once both the User Name and Server Name values have been entered or selected, the Composed User Name can be created. A sample field value rule is provided below.

```
import sailpoint.object.Application;
import sailpoint.tools.Util;
import sailpoint.pse.connector.MultiConnectorAdapter;
```

```
import sailpoint.pse.connector.MultiConnectorAdapter.MultiAppUtils;
import sailpoint.pse.connector.MultiConnectorAdapter.ObjectNameFormat;

if (application == void || application == null) {
    if (field != void && field != null && Util.isNotNullOrEmpty(field.getApplication()))
    {
        application = context.getObjectByName(Application.class, field.getApplication());
    } else {
        return null;
    }
}

if (application != null && Util.isNotNullOrEmpty(IIQ_multiUserName) &&
Util.isNotNullOrEmpty(IIQ_multiServerName)) {
    String formatName =
application.getStringAttributeValue(MultiConnectorAdapter.ARG_OBJECT_NAME_FORMAT);
    if (Util.isNotNullOrEmpty(formatName)) {
        formatName = "AT";
    }
    return MultiAppUtils.applyObjectNameTemplate(formatName, IIQ_multiUserName,
IIQ_multiServerName);
}
```

3.4 Credential Manager Integration

The Multi-Connector Adapter can be used in conjunction with the modules for PAM credential retrieval that are supported by IdentityIQ. It requires a small modification to the credential manager configuration for this to work. This option can be used instead of a credential rule, when a more static configuration is used.

Within the `Configuration` object “`CredentialConfiguration`”, one or more credential sources (typically PAM systems) can be configured. Within each entry, there is a generic configuration with information on how to connect to the credential source. Next, within each `CredentialSource` entry, there will be `CredentialAssociation` entries, which specify each credential attribute to retrieve for managed applications.

“Normal” applications will be listed by using the name of the application definition (Application object), but for systems managed by the Multi-Connector Adapter, the system identifier is used and prefixed with “MCATEMP-”, i.e a system identified as “LinuxMini1” to the Multi-Connector Adapter configurations, will have to be listed as **MCATEMP-LinuxMini1** in the `CredentialConfiguration` object to be recognized.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Configuration PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Configuration name="CredentialConfiguration">
    <Attributes>
        <Map>
            <entry key="sources">
                <value>
                    <List>
```

```

        <CredentialSource
credentialClass="sailpoint.pam.credential.WorkloadPrivilegeManagementCredentialManager
" name="WPMCredManager">
    <Attributes>
        <Map>
            <entry key="jwt" value="*****"/>
            <entry key="url"
value="https://acme.wpm.sailpoint.com/rest"/>
        </Map>
    </Attributes>
    <CredentialAssociation applicationName="MCATEMP-LinuxMini1"
attributeName="SudoUser" credentialAttributeName="userName">
        <Attributes>
            <Map>
                <entry key="credentialId" value="1234"/>
            </Map>
        </Attributes>
    </CredentialAssociation>
    <CredentialAssociation applicationName="MCATEMP-LinuxMini1"
attributeName="SudoUserPassword" credentialAttributeName="pwd">
        <Attributes>
            <Map>
                <entry key="credentialId" value="1234"/>
                <entry key="decodeBase64" value="true"/>
            </Map>
        </Attributes>
    </CredentialAssociation>
    </CredentialSource>
</List>
</value>
</entry>
</Map>
</Attributes>
</Configuration>

```

3.5 Auditing

Starting with version 2021.09.30-0002 (0.0.9), the connector supports auditing for failed aggregation and connection tests. For this to work, the XML document AuditConfig.xml that comes with the connector must be imported, via the console or via the **Global Settings → Import from file**.

With that document, the audit options will be enabled by default:

Multi-Connector Adapter Failed Aggregation	<input checked="" type="checkbox"/>
Multi-Connector Adapter Failed Connection Test	<input checked="" type="checkbox"/>

In addition, the attributes `auditTestConnectionFailure` and `multiAuditAggregationFailure` must be configured in the application definition (3.1) to enable auditing. This two-step configuration allows for enabling and disabling the auditing globally and per multi-application and feature.

4 Rule Examples

4.1 Multi Configuration Rule

The following rule demonstrates how entries from an LDAP server can be processed and used to create a configuration map.

```
import javax.naming.Context;
import javax.naming.directory.Attribute;
import javax.naming.directory.Attributes;
import javax.naming.ldap.LdapContext;
import javax.naming.ldap.InitialLdapContext;
import javax.naming.ldap.LdapName;
import javax.naming.directory.SearchControls;
import javax.naming.directory.SearchResult;
import javax.naming.NamingEnumeration;

Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldaps://ldap.acme.com:636/");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL,
"cn=sailpoint.admin,ou=special,ou=users,dc=acme,dc=com");
env.put(Context.SECURITY_CREDENTIALS, "*****");

LdapContext ctx = new InitialLdapContext(env, null);
ArrayList servers = new ArrayList();

LdapName name = new LdapName("ou=systems,dc=acme,dc=com");
SearchControls sc = new SearchControls();
sc.setSearchScope(SearchControls.SUBTREE_SCOPE);
NamingEnumeration<SearchResult> results = ctx.search(name, "(objectClass=ipHost)",
sc);
while (results.hasMore()) {
    SearchResult result = results.next();
    if (result != null) {
        Attributes attrs = result.getAttributes();
        Attribute cnAttr = attrs.get("cn");
        Attribute ipAttr = attrs.get("ipHostNumber");
        if (cnAttr != null && ipHostNumber != null) {
            Map server = new HashMap();
            server.put("multiHostIdentifier", cnAttr.get().toString());
            server.put("host", ipAttr.get().toString());
            server.put("SshPort", "22");
            servers.add(server);
        }
    }
}
return servers;
```

4.2 Multi Credential Rule

The following example rule makes use of the IdentityIQ support for PAM credential rotation and live retrieval from a PAM system.

```
import sailpoint.credential.CredentialRetriever;
import sailpoint.pse.connector.MultiConnectorAdapter;
import sailpoint.pse.connector.MultiConnectorAdapter.MultiAppUtils;
import sailpoint.tools.Util;

try {
    if (application != null && server != null) {
        String connectorName =
application.getStringAttributeValue(MultiConnectorAdapter.ARG_APP_CONNECTOR);
        if (Util.isNotNullOrEmpty(connectorName)) {
            Application app = MultiAppUtils.createServerApplication(context, application,
connectorName, server);
            if (app != null) {
                CredentialRetriever cr = new CredentialRetriever(context);
                if (cr.hasConfiguration(app)) {
                    cr.updateCredentials(app);
                    Map result = new HashMap();
                    result.put("SudoUser", app.getStringAttributeValue("SudoUser"));
                    result.put("SudoUserPassword",
app.getStringAttributeValue("SudoUserPassword"));
                    return result;
                }
            }
        }
    }
} catch (Exception e) {
    log.error(e);
}
return null;
```

In the above example, first, the connector logic to create a derived application, is used to compose an application definition. Then, a `CredentialRetriever` class is instantiated and used to determine whether an entry exists in the `CredentialConfiguration` for the application definition. If so, the `CredentialRetriever` is used to fetch the credentials. The updated application definition is then used to extract the username and password attributes and place them in the return map.

4.3 Before Provisioning Rule

An example of a Before Provisioning Rule that inspects the plan and corrects the relations between accounts and requested entitlements is given below:

```
import sailpoint.pse.connector.MultiConnectorAdapter.MultiAppUtils;

/*
 * The following code will inspect the provisioning plan and make last-minute updates
 * to try and ensure that the plan gets provisioned correctly.
 *
 * If a plan contains a entitlementment that is related to a different target than the account or
```



```
* object name, the connector will fail. The method call below is used to do that, but with the
* inputs 'ignoreIncorrect' and 'autoCorrect' both set to false.
*
* To prevent that from failing, the following code be configured and used:
*
* Set 'ignoreIncorrect' to true to ignore any mismatch --> not further processing, silently
ignored.
*
* Set 'autoCorrect' to true to try and correct the plan. This will re-evaluate the provisioning
policies
* after corrections have been made. Corrections include cloning requests for attributes so they
* will be sent to the correct server. For this setting to work, 'ignoreIncorrect' must be false!
*/

// Ignore Incorrect
boolean ignoreIncorrect = false;

// Auto-Correct
boolean autoCorrect = true;

if (plan != null) {
    log.debug("Plan before autocorrect: " + plan.toXml());
    MultiAppUtils.checkPlanPreProvisioning(context, application, plan, ignoreIncorrect,
autoCorrect);
}
if (plan != null) {
    log.debug("Plan after autocorrect: " + plan.toXml());
}
```

5 Support Plugin

The Multi-Connector Adapter Support Plugin can be installed next to the connector to support the operations of the connector. It is supported on IdentityIQ 8.0 and up and connector version 2021.09.08-0003 and higher. The list of features will be growing over time.

Features:

- Store aggregation results (connector version 2021.09.08-0003 and higher).
- Background connection test task.
- Auto-Correct LCM Provisioning Workflow Wrapper

The connector will automatically detect the existence of the plugin and if the plugin is installed and active, make use of it.

5.1 Store Aggregation Results

The plugin comes with two tables for storing aggregation results:

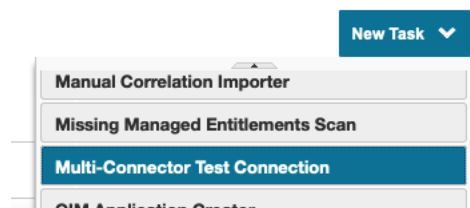
- One table is used to store the last successful aggregation of managed target systems per multi-application, per object type.
- The second table is used to store the list of target systems that failed during the last aggregation per multi-application, per object type.

This feature replaces the storage of aggregation results in the application XML, which could otherwise become very large and inefficient for large environments.

5.2 Background Connection Test Task

The plugin includes a task executor and task template to perform connection testing on one or more Multi-Connector Applications and their targets.

A new task can be set up by selecting the **Multi-Connector Test Connection** option from the **New Task** menu.



The task can use threads or partitions to efficiently perform simultaneous tests on several targets. For partitioned tests, an additional module needs to be installed, the Plugin Request Executor (see chapter 6).

The task has the following options.

Option	Description
Applications to scan*	One or more multi-connector applications. These can be different types. Note: <i>it is possible to select non-multi-connector applications. These will be ignored by the task and not tested.</i>
Only check systems that have previously failed.	If this option is used, the list of target systems that have failed the test during the last run will be used. This is useful when a daily test is run for all targets and several times per day only the failed ones.
Ignore failing systems after a number of days (1 or more, 0 to never ignore)	This option is used in conjunction with the previous. If systems have failed, they will not be included if they have not been tested successfully for more than the specified number of days. If the value is set to 0, no system will be ignored.
Include Account Aggregation Test.	If enabled, the test will include a limited account aggregation from each of the target systems. This is similar to the preview button on an application's account schema.
Include Group Aggregation Test.	If enabled, the test will include a limited group aggregation from each of the target systems, and for each of the group types. This is like the preview button on an application's group schemas.

Option	Description
Aggregation Test Object Count.	This setting applies to the previous two options. If any of these two options is enabled, the aggregation test will get this many objects from the target system, before closing the connection.
Test Connection Retries.	<p>If this option is set and 1 or higher, targets that fail the test can be retried. The maximum number of retries is 10. Only errors listed as retryable errors in the application definition will be a reason to retry (see 2.2.3).</p> <p>Before each retry a delay is added of $n * 100$ milliseconds, where n is the number of tries. If the option "Random Delay between Targets" is enabled, the delay will be random.</p>
Random Delay between Targets.	<p>Enable this option to add a random delay before testing each target. The default range is between 0 and 1000 milliseconds but can be controlled by the option "Maximum Random Delay (seconds)."</p> <p>The purpose of this option is to prevent or debug issues with concurrent tests overloading infrastructure systems like PAM servers providing passwords.</p>
Maximum Random Delay (seconds).	<p>This option controls the range for the option "Random Delay between Targets." The default and minimum value is 1, corresponding to 1000 milliseconds. The maximum value is 180 seconds, corresponding to a maximum delay of 180,000 milliseconds, or 3 minutes.</p> <p>The recommendation is to keep the number as low as possible and increase only if needed.</p>
Pre-Test Rule	A rule can be selected that is executed for each of the targets to be tested (5.2.1).

Option	Description
Pre-Test Rule Arguments	<p>A string that can contain multiple arguments that will be used as input for the Pre-Test Rule, in addition to standard variables. The arguments are to be formatted as <code>name=value</code> and each argument is separated by a pipe character (<code>' '</code>). For example:</p> <pre>verbose=true mailto=admin@example.org</pre> <p>Note that specified arguments may be overridden by standard variable values (<code>log</code>, <code>purpose</code>, <code>server</code>, etc.).</p>
Post-Test Rule	A rule can be selected that is executed for each of the targets after testing (5.2.2).
Post-Test Rule Arguments	<p>A string that can contain multiple arguments that will be used as input for the Post-Test Rule, in addition to standard variables. The arguments are to be formatted as <code>name=value</code> and each argument is separated by a pipe character (<code>' '</code>). For example:</p> <pre>verbose=true mailto=admin@example.org</pre> <p>Note that specified arguments may be overridden by standard variable values (<code>log</code>, <code>purpose</code>, <code>server</code>, etc.).</p>
Aggregate After Successful Test	If a test (including aggregation tests, if enabled) is successful, perform a full aggregation of the target.
Aggregate Dry Run	If 'Aggregate After Successful Test' is enabled, selecting this option, will make the aggregation a dry run: all groups and accounts will be read and counted, but nothing will be persisted in IdentityIQ.

Option	Description
Aggregation Timeout	<p>If a full aggregation or dry run is performed, a timeout can be configured in seconds. This timeout is applied to each target and each schema type that is aggregated. If the timeout is reached, the aggregation or dry run is aborted and will be considered failed.</p> <p>The purpose of this is to identify targets that would take an unusual amount of time to aggregate from.</p>
Only create links if they can be correlated to an existing identity.	<p>This is a standard aggregation option. It ensures that an account is only persisted if it can be correlated to an existing identity. Uncorrelatable accounts will not be persisted.</p> <p>This option has no effect on a dry run.</p>
Disable optimization of unchanged accounts	<p>This is a standard aggregation option. If selected, all accounts will be refreshed, even if they are unchanged since a previous aggregation.</p> <p>This option has no effect on a dry run.</p>
Disable Native Change Detection	<p>During an initial aggregation, it may be desired to disable Native Change Detection (default: true), to avoid false positives and set a baseline for a newly on-boarded system. Disable this option to record native changes.</p> <p>Note: this option is only effective on a non-dry-run aggregation.</p>
Group Aggregation Refresh Rule	<p>The selected rule will be run if a non-dry-run aggregation is performed to allow updating the <code>ManagedAttribute</code> object for each aggregated group. This is a standard rule type for Account Group Aggregation tasks.</p>

Option	Description
Number of Threads (default 1, max 64)	The number of threads for connection testing on a single IdentityIQ server. The number is 1 by default, and the maximum is 64. The recommendation is to choose the number equals to or less than the number of CPU cores on the IdentityIQ servers.
Enable Partitioning	Enable partitioned execution. This requires the Plugin Request Executor (see chapter 6). If this module is not available, the execution will make use of threading, even if this option is enabled. If the module is available, the number of threads is ignored. The number of partitions is determined based on the number of active request servers, multiplied by the number (<code>maxThreads</code>) configured on the <code>RequestDefinition</code> object named <code>Multi-Connector Adapter Test Connection Request</code> .
Test Purpose	This is a parameter that is provided to the configuration rule in the Multi-Connector Adapter, if used. The parameter can be used to determine the purpose of execution. If nothing is provided, the value <code>"TEST_CONNECTION"</code> is used.
Audit Instance Tag	To support searching or grouping for specific tasks or specific runs of the task in audit records for successful and failed tests, a string can be provided that will be parsed using Apache Velocity to produce a unique string for each task run. See 5.5.1 .
Throw an exception if no targets defined	If no targets are found, by default, the test connection task will end successfully and report a warning. If this option is enabled, the task will end in an error by throwing an exception.

MultiLinux Test Connection Options

Applications to scan*	?	<div> <div></div> <div>MultiLinux</div> </div>
Only check systems that have previously failed.	?	<input type="checkbox"/>
Ignore failing systems after a number of days (1 or more, 0 to never ignore)	?	<input type="text"/>
Include Account Aggregation Test.	?	<input checked="" type="checkbox"/>
Include Group Aggregation Test.	?	<input checked="" type="checkbox"/>
Aggregation Test Object Count.	?	<input type="text" value="10"/>
Test Connection Retries.	?	<input type="text" value="1"/>
Random Delay between Targets	?	<input type="checkbox"/>
Maximum Random Delay (seconds).	?	<input type="text" value="1"/>
Pre-Test Rule	?	-- Select an Object --
Pre-Test Rule Arguments	?	<input type="text"/>
Post-Test Rule	?	-- Select an Object --
Post-Test Rule Arguments	?	<input type="text"/>
Aggregate After Successful Test	?	<input checked="" type="checkbox"/>
Aggregate Dry Run	?	<input type="checkbox"/>
Aggregation Timeout	?	<input type="text" value="30"/>
Only create links if they can be correlated to an existing identity.	?	<input type="checkbox"/>
Disable optimization of unchanged accounts	?	<input type="checkbox"/>
Disable Native Change Detection	?	<input checked="" type="checkbox"/>
Group Aggregation Refresh Rule	?	-- Select an Object --
Number of Threads (default 1, max 64)	?	<input type="text" value="8"/>
Enable Partitioning	?	<input checked="" type="checkbox"/>
Test Purpose	?	<input type="text" value="Enabled"/>
Audit Instance Tag	?	<input type="text" value="Linux Verification \$fulltimestamp (\$runs)"/>
Throw an exception if no targets defined	?	<input checked="" type="checkbox"/>

5.2.1 Pre-Test Rule

The post-test rule, if selected, is run for each target just after the connector is closed. It allows for action after successful or unsuccessful tests, for example to notify an administrator.

The rule will get, apart from the default `log` and `context` variable, the following inputs:

Variable	Type	Description
<code>applicationName</code>	String	The application name of the base Multi-Connector Application.
<code>purpose</code>	String	The purpose entered in the task definition.
<code>targetName</code>	String	The identifier for the target system to test.
<code>targetConfiguration</code>	Map<String,Object>	A map with the target-specific attributes.

In addition, the Pre-Test Rule Arguments may provide additional, static inputs. There is no output expected. Exceptions are caught and logged.

5.2.2 Post-Test Rule

The pre-test rule, if selected, is run for each target just before the connector gets instantiated. It allows for some last-minute checks, e.g., to put a Credential Association in place based on information from the Custom object or CMDB.

The rule will get, apart from the default `log` and `context` variable, the following inputs:

Variable	Type	Description
<code>applicationName</code>	String	The application name of the base Multi-Connector Application.
<code>purpose</code>	String	The purpose entered in the task definition.

Variable	Type	Description
targetName	String	The identifier for the target system to test.
targetConfiguration	Map<String,Object>	A map with the target-specific attributes.
success	boolean	True if successful, false if not.
errors	List<String>	A list of errors. Null or empty on success.

In addition, the Pre-Test Rule Arguments may provide additional, static inputs.

There is no output expected. Exceptions are caught and logged.

5.2.3 Connection Test Scenarios

The Connection Test Task is meant to simplify and help automate on-boarding and off-boarding of targets. Below, a few example scenarios are listed to illustrate how this task can help.

An assumption for these scenarios is that the data for the targets is managed in a way that IdentityIQ can update information. In real-life customer scenarios, the Entitlement Catalog has been used to store information about the managed targets and their metadata. The Configuration Rule would search for the relevant `ManagedAttribute` objects and return these.

5.2.3.1 Testing New Targets for On-Boarding

For on-boarding of new targets, the targets that have a status 'New', need to be tested before they will be enabled for use in aggregation and provisioning. The purpose of the test is to verify whether everything is in place and where possible, correct the information:

- Verify whether the credential cycling is set up correctly. If not, update the configuration to point to the correct PAM server instance, container and object, as retrieved from the metadata. (Pre-Test Rule: 5.2.1)
- Verify whether the target can be reached by performing a basic connection test.
- Verify that account and group aggregation work by reading a small number of records, very similar to the preview feature of an application definition.

- If all goes well so far, a full aggregation or dry run aggregation can be performed, possibly with a configured timeout to identify targets that would take too much time to aggregate.
- When the test completes for a target, the target can be moved to the next stage. Depending on the customer's preferences, the target can be enabled automatically, or someone will have to manually enable the hosts that have status 'Verified', or a workflow can be started allowing the responsible persons to perform a last verification and provide their approval before the target is enabled. (Post-Test Rule: 5.2.2)

A report can be extracted with the status of successful and failed connection tests for further analysis.

5.2.3.2 Testing Enabled Targets

A periodic run of the task can help to identify targets that do no longer respond. There could be several reasons for a target not to respond. So, before automatically disabling a target, it would be best to notify the persons responsible for managing the system.

The test connection task can as an example be set up as follows:

- The task will select targets that have the status 'Enabled', by specifying this status as the 'purpose'.
- For each enabled targets, the task will simply perform a basic connection test.
- If a target fails, the task will create an audit record.
- For failed targets, the Post-Test Rule (5.2.2) can:
 - Notify the operator by email.
 - Verify how often the target has failed or how much time has passed since the last successful test. If a threshold is reached, the target can be disabled automatically.

5.3 Partitioned Group Aggregation

From IdentityIQ 8.3 and up, the group aggregation task supports partitioning. This partitioning, however, is not the same as for account aggregation. It will first read all information and then starts processing in partitions.

As that is far from ideal for large environment, a real partitioned group aggregation is needed for the Multi-Connector Adapter. This is implemented using the **Multi-Connector Partitioned Account Group Aggregation** task template. This task will ask

the connector to generate partitions. It can also be used with connectors other than the Multi-Connector Adapter, but most connectors do not yet support group partitioning. In that case a dummy partition is created for a “normal” group aggregation.

The task has the following options:

Option	Display Name	Description
<code>applications</code>	Select applications to scan	The applications to aggregate.
<code>deltaAggregation</code>	Enable Delta Aggregation	Use delta aggregation. The use of this option is not recommended for the group aggregation task as it will block the option Detect deleted groups .
<code>checkDeleted</code>	Detect deleted account groups	If enabled, the task will generate partitions to clean up groups per application, per group schema.
<code>checkDeletedThreshold</code>	Maximum deleted groups	The maximum number of deleted groups. This is evaluated per application. So, if this is set to 100, application A has 234 deleted accounts and application B has 53, then only deleted accounts will be processed for application B, but not for A.
<code>descriptionLocale</code>	Automatically promote descriptions to this locale	Automatically set descriptions on the aggregated account groups from the selected

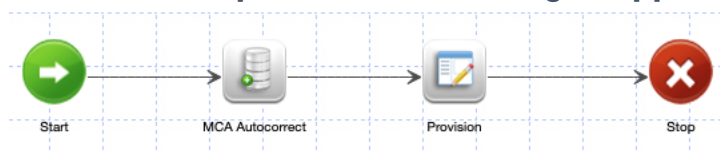
Option	Display Name	Description
descriptionAttribute	Description attribute (default "description")	attribute in the selected locale (e.g. en_US)
accountGroupRefreshRule	Group Aggregation Refresh Rule	This rule is used to set the owner or modify the account group when it is created or refreshed.
promoteClassifications	Promote Classifications	Automatically set the classification of the account groups, from the selected attribute.
classificationAttribute	Classification Attribute	

5.4 Auto-Correct LCM Provisioning Workflow Wrapper

To handle incorrect account selections, a rule can be used (4.3). This rule is executed just before the provisioning plan is handed to the connector. This approach has a few downsides:

- The plan as sent to the connector may differ from what is recorded in the identity request, potentially leading to confusion or incorrect validation of the provisioned data.
- The account create provisioning policy **must** be fully automated and no user interaction should be required. If any user interaction is needed to fill in attributes, the rule will fail.

An improved approach is to fix up incorrect account selections before submitting the plan to the provisioning workflow. This is done in a wrapper workflow: **Multi-Connector Adapter LCM Provisioning Wrapper**.



The **Start** and **Stop** steps are placeholder steps and do nothing. The step **MCA Autocorrect** uses library methods in the plugin to check for any corrections needed and update the plan:

- Check for entitlements assigned to the wrong account.
- Check for accounts that need to be created.
- Leave any requests for non-MCA applications or for MCA applications that have account merging enabled unchanged.

The step **Provision** will call the out-of-the-box **LCM Provisioning** workflow.

All options that are normally configured on the LCM Provisioning workflow can be configured using the option on the **Multi-Connector Adapter LCM Provisioning Wrapper** workflow and will be passed to **LCM Provisioning**.

To use the **Multi-Connector Adapter LCM Provisioning Wrapper** workflow, it should be selected in the Lifecycle Manager configuration screen under **Business Processes**.

Lifecycle Manager

Configure Business Processes Identity Provisioning Policies	
Action	Business Process
Request Access	? Multi-Connector Adapter LCM Provisioning Wrapper ▾
Manage Accounts	? LCM Provisioning ▾
Unlock User Account	? LCM Provisioning ▾
Manage Passwords	? LCM Manage Passwords ▾
Edit Identity	? LCM Create and Update ▾
Create Identity	? LCM Create and Update ▾
Self-service Registration	? LCM Registration ▾
Batch Request Access	? Multi-Connector Adapter LCM Provisioning Wrapper ▾
Batch Manage Accounts	? LCM Provisioning ▾
Batch Manage Passwords	? LCM Manage Passwords ▾
Batch Edit Identity	? LCM Create and Update ▾
Batch Create Identity	? LCM Create and Update ▾

5.4.1 Alternative Implementation Options

Many customers use a modified version (copy) of the **LCM Provisioning** workflow. In that case, the functionality of the **Multi-Connector Adapter LCM Provisioning Wrapper** workflow can be used in two ways:

- Customers can make a copy of the **Multi-Connector Adapter LCM Provisioning Wrapper** workflow and update the **Provision** step to call their own version of **LCM Provisioning**.
- Customers can copy the **Provision** step of the **Multi-Connector Adapter LCM Provisioning Wrapper** workflow and inject that into their modified version of the **LCM Provisioning** workflow, before the

5.5 Auditing

The plugin comes with the following audit events, which will be activated out of the box.

Event	Description
Multi-Connector Adapter Task Failed Connection Test	This event is generated by the plugin-based connection test task (5.2) for every target system that does not respond (correctly).
Multi-Connector Adapter Task Successful Connection Test	This event is generated by the plugin-based connection test task (5.2) for every target system responds correctly.

5.5.1 Audit Instance Tag

The Audit Instance Tag is a string that is parsed using Apache Velocity and will be added in the audit events produced for every successful or failed connection test as the `instance` variable. The inputs for the “tag” consist of all attributes of the task, plus the variables listed in the table below.

Variable	Type	Description
<code>\$fulltimestamp</code>	String	The start date and time of the task execution formatted as <code>yyyy-MM-dd HH:mm:ss.SSS</code> .
<code>\$timestamp</code>	String	The start date and time of the task execution formatted as <code>yyyy-MM-dd HH:mm</code> .

Variable	Type	Description
\$datestamp	String	The start date of the task execution formatted as yyyy-MM-dd.
\$runs	Integer	The number of times the task has been run.

An example “tag” could look like this:

```
MySQL Verification $fulltimestamp ($runs)
```

This could produce the following output:

```
MySQL Verification 2022-04-15 08:45:18.087 (123)
```

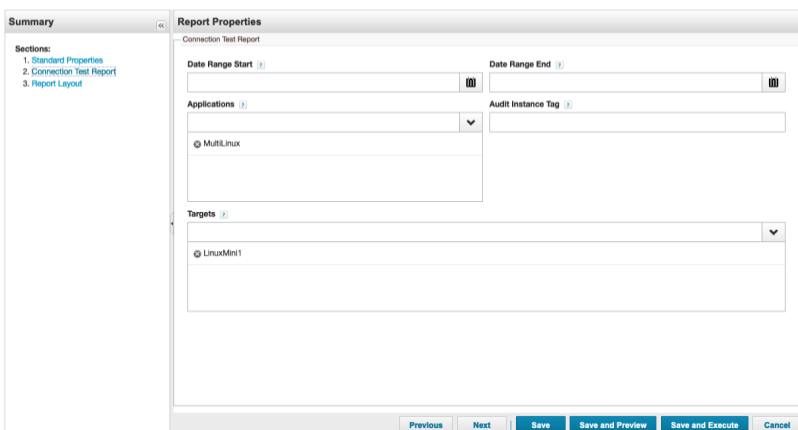
5.6 Reporting

The reports provided by the plugin are described below.

5.6.1 Multi-Connector Adapter Support Plugin Test Results Report

A report is included that lists the results of connection tests executed via the connection test task. The results included in the report can be filtered by a start and end data, by application, by the Audit Instance Tag (5.5.1) and by target. When applications are selected, the targets are limited to the those belonging to the selected applications.

[Edit Report](#)



The report can contain a column with ‘Potential Causes’ for errors that occur during connection testing. A Configuration object is used to maintain error messages and

their potential causes. The Configuration object has a section named “errors” that contains a map of error messages and for each message a list of potential causes. Each error can be specified as a plain string, which will be evaluated as a case-insensitive substring of the error message, or as a regular expression. Regular expressions must be prefixed with `{{REGEX}}`.

```
<Attributes>
  <Map>
    <entry key="errors">
      <value>
        <Map>
          <entry key="{{REGEX}}^.*Access denied for user '\w+'@'\w+'
(using password: YES).*$">
            <value>
              <List>
                <String>Service Account Credentials</String>
                <String>Login Host Restrictions</String>
              </List>
            </value>
          </entry>
          ...
          <entry key="Allowed authentication methods on UNIX host:">
            <value>
              <List>
                <String>Invalid Authentication Method</String>
              </List>
            </value>
          </entry>
          ...
```

The Configuration object also contains a map with suggested actions for each of the potential causes. The causes are case-sensitive. Some are generic, some are specific to a certain target connector type.

```
...
    <entry key="suggestions">
      <value>
        <Map>
          <entry key="Credential Rotation" value="Verify that the
credential rotation for the target is configured correctly, the credential source
(PAM) is reachable, the credentials are identified correctly, accessible and are in
the correct location." />
          <entry key="DNS Incorrect Hostname" value="Verify that the
specified hostname and domain are correct." />
          <entry key="Firewall" value="If the firewall is actively
rejecting traffic, a 'connection refused' can be returned. If the firewall ignores
disallowed traffic, a timeout may occur. Check all firewall between client and server,
including the host firewalls on the client and server hosts." />
          ...
```

A Configuration object is used instead of a Custom object, so merging can be used. Customers can inject their own environment specific error messages by adding their own messages in another “merge” object.

6 Plugin Request Executor

This is a special request executor adapter that can be used to execute request executors bundled in plugins. It is required for the partitioned execution of tasks bundled in a plugin.

The module is provided as a jar file (`plugin-request-executor.jar` or `plugin-request-executor-<version>.jar`) that must be installed under `WEB-INF/lib`, relative to the installation folder of IdentityIQ. IdentityIQ must be restarted for the module to work. The module must be installed on all request hosts.

In order to use a request executor that is bundled in a plugin, the executor must be set to `sailpoint.pse.request.PluginRequestExecutor`. In the attributes map, the name of the plugin must be provided using the attribute `pluginName`. The executor class must be configured using the attribute `pluginRequestExecutor`.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE RequestDefinition PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<RequestDefinition executor="sailpoint.pse.request.PluginRequestExecutor" name="Multi-
Connector Adapter Test Connection Request">
  <Attributes>
    <Map>
      <entry key="hostSpecific" value="true"/>
      <entry key="maxThreads" value="2"/>
      <entry key="pluginName" value="multi_connector_adapter_support_plugin"/>
      <entry key="pluginRequestExecutor"
value="com.sailpoint.pse.plugin.multiconnectoradaptersupportplugin.TestConnectionReque
stExecutor"/>
    </Map>
  </Attributes>
</RequestDefinition>
```

The attribute `maxThreads` is used to configure how many partitions will be run simultaneously on each IdentityIQ request server and to calculate the number of partitions to generate.

7 Known Limitations

7.1 Cloud Gateway

The use with a Cloud Gateway has not been tested!

If a cloud gateway is used, the cloud gateway must be able to reach all systems that are configured for the selected application. It is not possible to have a few systems behind the cloud gateway while others are not or reachable only through another cloud gateway. In that case, if there are multiple locations, each with their own cloud gateway, a Multi-Connector Adapter application must be set up per gateway.

If a cloud gateway is used as the proxy for the Multi-Connector Adapter:

- The connector will not be able to perform identity checks. This means that any optimization for provisioning (see example Before Provisioning Rule: 4.3) will have to run on IdentityIQ, but not on the cloud gateway.
- The use of the MCA Support Plugin will not be possible.
- Audit information will not be generated.
- If it not possible to use simulation mode.

7.2 Delta Aggregation

Currently, the connector adapter will not support delta aggregation from systems managed by the application. If delta aggregation is selected, the delta flag is not passed on to the real connector.

Delta aggregation is used by the adapter internally to allow only a specified, limited number of systems to be aggregated. For example, when during the last full aggregation, a few systems could not be aggregated from, they can quickly be retried using a delta aggregation task.

8 Frequently Asked Questions

8.1 Which application types can be used with the Multi-Connector Adapter?

Theoretically, any connector type can be used with the Multi-Connector Adapter. Several systems, mostly unix flavors, windows, and database host types, have already been connected successfully. Others need to be tested for full confirmation. If it is possible, it does not mean that it makes sense to do it. For example, for Active Directory, the out-of-the-box connector provides features to connect multiple forests and domains, which removes the need to create a multi-version of that. The out-of-the-box connector provides features to link accounts and groups between trusted domains, which will be harder with the Multi-Connector Adapter.

8.2 Does the adapter run on a separate host, or does it need to be installed on every IdentityIQ instance?

The adapter is installed like any other connector. It has Java classes, configuration pages (xhtml) and one or more entries in the connector registry (XML). It needs to be installed on any host that deals with aggregation and/or provisioning, including those where one may perform “test connection” or “preview” on the account or group schemas. Usually, the connector code and artifacts would be included in the build process and indeed installed on every host.

8.3 Are there any specific deployment considerations (high-availability, network latency, etc.)?

There are no other deployment considerations compared to the normal connectors for the target systems.

Installation on the Cloud Gateway has not been validated, yet, though.

8.4 Are there data volume limits or performance considerations?

So far, we have tested with up to 1000 target systems for aggregation in a lab situation. This works smoothly and fast.

There is no technical limit to the number of hosts, but there may be multiple practical reasons why a company may want to divide the list of systems:

- Run-time: just too many systems taking too much time for a single aggregation task
- Regional separation: group systems by their geographical location (country, continent, time zone)
- Responsibility: group by operations team, department, responsible managers, etc.
- Common configuration settings, such that the parameters per system may be limited to just connectivity and credentials.
- Dynamic versus static sets of servers: some systems may be permanent, some may come and go; keep the dynamic list separate from the static list, vary the retry options, etc.

The Multi-Connector Adapter provides features to optimize the aggregation runtime: standard optimization can be used and with partitioning, multiple hosts can be processed in parallel. The usable number of partitions depends on the number of CPU cores (see the Partitioning Best Practices on Compass²).

The Multi-Connector Adapter does *not* support real delta aggregation, but delta aggregation can be used to only aggregate specific hosts, hosts that have failed during the last aggregation or have not been aggregated successfully over the past X days (X is configurable).

Large numbers of hosts and large amounts of data being aggregated will affect the runtime of aggregation tasks. When aggregations are starting to run too long, it should be considered to scale up the environment by adding more CPUs or more hosts, allowing more partitions to be executed in parallel.

² <https://community.sailpoint.com/t5/Other-Documents/Partitioning-Best-Practices/tap/74964>

The large volume of the data may also require the sizing of the database to be adjusted. The general recommendation to reduce application schemas to the absolute minimum also applies to the Multi-Connector Adapter and is likely even more important here, to reduce the volume of the data being stored.

8.5 How to Avoid the “Fat Identity” Problem

With hundreds or thousands of hosts, there is a risk of running into the so called “fat identity problem”, a problem where aggregation and refresh performance may degrade if lots of identities each have a lot of accounts. The recommendation is to try and keep the number of accounts per identity below 100. In practice, 200 for some identities may still be workable and in testing we’ve had several identities with over 1000 accounts.

To prevent the creation of these ‘fat identities’, it is recommended to treat services and standard accounts that exist on every system (like ‘root’ and ‘nobody’ on unix systems or ‘Administrator’ on Windows) as individual identities and not to correlate them to a single service-identity. The Multi-Connector Adapter is targeted at situations where the company may have hundreds or thousands of systems, but (real, human) users typically have accounts on a subset of these systems.

In case some people do have 100s or 1000s of personal accounts, one should think about a different correlation strategy for such accounts. Instead of directly correlating such accounts to the ‘holder’ of these accounts, an ‘ownership’ or ‘administrator’ relation could be more appropriate.

8.6 Which logging mechanism does the Multi-Connector Adapter use?

As the Multi-Connector Adapter works like a normal connector, it also uses the same, out-of-the-box logging mechanisms. To debug the connector, one should add the following lines to the `log4j2.properties` file:

```
logger.multiconnectoradapter.name=sailpoint.pse.connector.MultiConnectorAdapter
logger.multiconnectoradapter.level=debug
```

For more verbose output, set the level to `trace`.