

IBM Security Verify Information Queue User's Guide

Contents

ISIQ User Authentication	4
Logging In.....	4
Select a Product.....	5
Configuring ISIM	6
Note on ISIQ Navigation.....	10
Verifying ISIM Data Loading.....	11
Configuring IGI.....	13
Subscriptions	14
Connectors and Topics	18
Reprocessing Topics	20
Verifying Product Registration.....	21
Transformers	22
ISIM-to-IGI Integration	23
External Applications	24
Metrics and Alerts	29
Group Management.....	32
Custom Transformations.....	35
Logging Out.....	37
Appendix A: IGI Customizations for ISIQ.....	38
Define adequate temporary table space size	38
Review ISIM customizations	38
Enable the EVENT_OUT_USER table	38
Update the USER_ERC attribute mapping to set the user DN	39
Update the Create Account rule flow.....	40
Update the Create Org. Unit rule flow	41
Update the Create User rule flow	41
Update the Modify OrgUnit rule flow	42
Set Group DN to entitlement	43

Optional IGI customizations	44
Integrate ISIM orphan accounts into IGI as orphan accounts	44
Allow the account user ID to be changed for a service account.....	46
Support multiple user sources for IGI.....	47
How to create a new rule	47
How to replace an existing rule with a new rule	47
Appendix B: ISIM Customizations for ISIQ	48
Processing ISIM deletes.....	48
Resetting the Change Log Offset.....	48
Appendix C: Mapping Entities and Events between ISIM and IGI	49
Entity-to-Entity Mapping.....	49
Entity-Key Field Mapping.....	49
Events Fulfilled for IGI-to-ISIM Integration.....	50
Updating ERC Status in the IGI OUT Events Queue	51
Interpreting the Justification Field	51
Creating Custom Persons in ISIM	51
Ignoring IGI Account Events from a Specific User	52
Supported Operations on ISIM and IGI Entities.....	52
Appendix D: Updating Subscriptions.....	54
Appendix E: Custom ISIM-to-IGI Transformations	56
How to Customize Transformations for Accounts.....	57
How to Customize Transformations for Users	58
How to Support Custom Persons and BPPersons.....	61
How to Support a Custom Service.....	62
Find Object Profile Name for Service, Group, and Account	62
Update txdef.json with the new profile names	67
How to Skip OU Synchronization.....	70
Additional ISIM Customization Options.....	74
Support Auxiliary Object Classes	75
Configure LDAP Connection Pool	75
Configure LDAP Search Limit and Lookahead Values	76
Configure LDAP Poll Delay Interval	76
Map Embedded ISIM Attribute to IGI Attribute	76

Block Modify Account Events to IGI	77
Reduce ISIM Database Queries	77
Exclude or Include Services	78
Implement a Naming Attribute Map	78
Implement a Profile Topic Map	79
Restart the ISIQ Stacks	79
Update the IGI-to-ISIM Subscription	79
Appendix F: Removing ISIM Data in IGI	80
Log in as Admin User	80
Remove Applications	80
Remove Account Configurations	81
Remove Rights Lookups	81
Remove Users	81
Remove Organizational Units	81
Remove Events	82
Reloading ISIM Data	82
Appendix G: ISIM Integration with Cloud Identity	84
Appendix H: Optional ISIQ Tools	89
ISIM-to-IGI Validation Tool	89
ISIGADI-to-ISIQ Migration Tool	89
Setting Initial Offsets for ISIM and IGI Source Topics	89

IBM® Security Verify Information Queue, which uses the acronym “ISIQ”, is a cross-product integrator that leverages Kafka technology and a publish/subscribe model to integrate data between IBM Security products.

This User’s Guide explains ISIQ concepts and terminology and provides an example with screen captures to help you get started.

ISIQ User Authentication

After downloading ISIQ from IBM Cloud Container Registry, your system administrator must configure an OpenID Connect (OIDC) provider to authenticate users. For more information about OIDC configuration, see the “OpenID” section of the [ISIQ Deployment Guide](#).

Once your ISIQ docker swarm starts, you can log in to ISIQ’s web interface. For many OIDC providers, your email ID is your user ID. During ISIQ user authentication, the name portion of a supplied email ID gets parsed out and assigned as the subscriber ID. For instance, if your email ID is johndoe@acme.com, your ISIQ subscriber ID is “johndoe”. Alternatively, if IBM Security Verify Governance (referred to as “ISVG” but previously known as IBM Security Identity Governance and Intelligence with the “IGI” acronym) was configured as ISIQ’s OIDC provider and you log in with an IGI user called “admin”, your ISIQ subscriber ID is “admin”.

Instead of logging in as an individual, you can log in as a member of a group. If multiple people in your organization need to work with a common set of ISIQ products and subscriptions, it makes sense to use the group feature (for more information, see “Group Management” later in this guide). By doing so, each group member has access to all ISIQ entities belonging to the group. Any product or subscription defined by a group member gets assigned an ISIQ subscriber ID that is a hash representation of the group name.

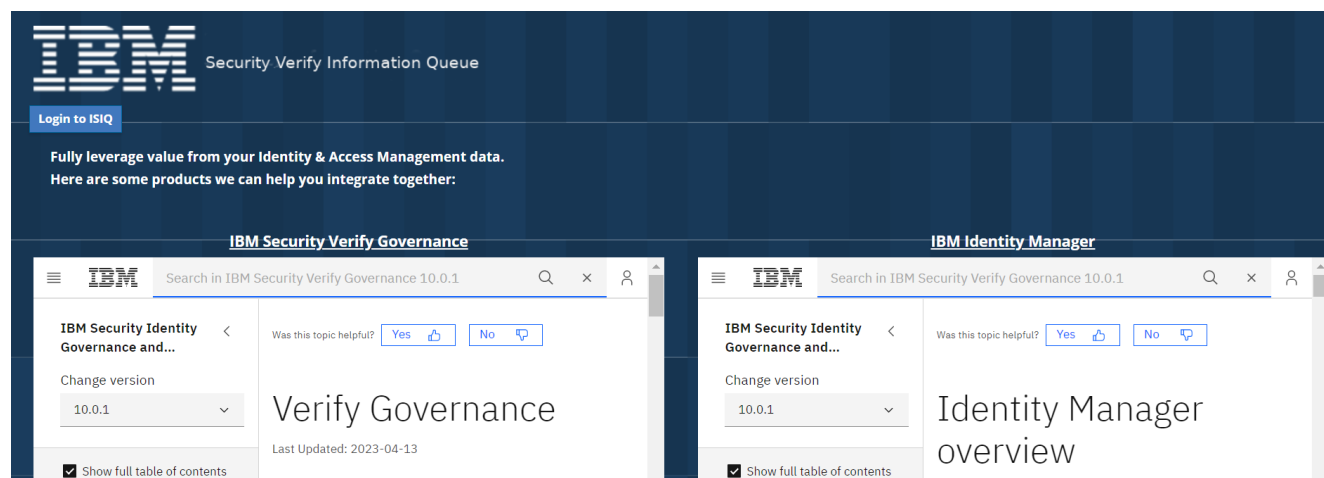
The subscriber ID is important in ISIQ. It’s used as a prefix when naming connectors, topics, and sources to subscribe to. The subscriber ID functions as an owner ID since the subscriber prefix indicates who owns the ISIQ entity. If you don’t use the group feature, your access is restricted to entities prefixed with your individual subscriber ID.

Logging In

The ISIQ login URL is:

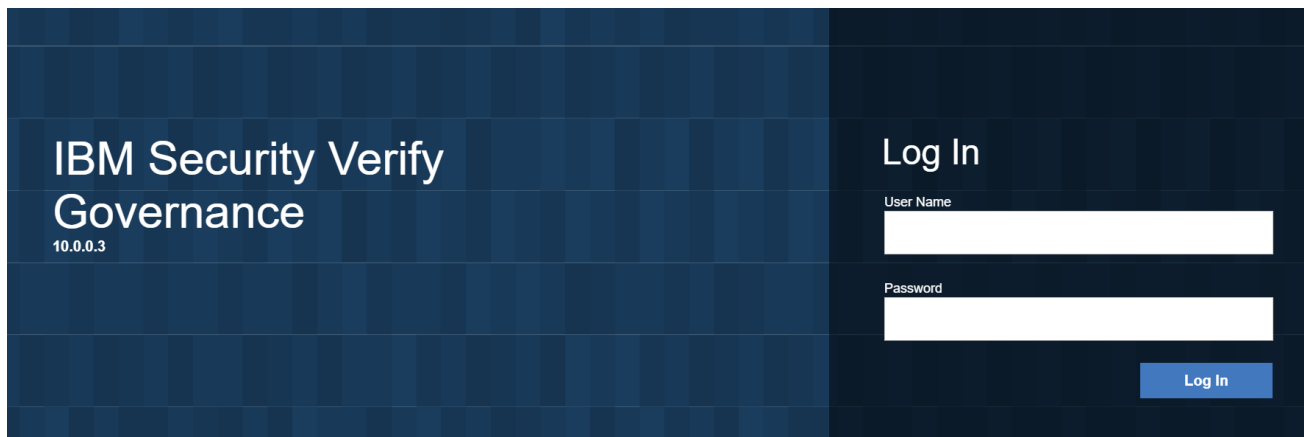
`https://<your-ISIQ-hostname>/#!/login`

After you enter this URL in a browser, you should see the following page:



Click the “Login to ISIQ” button to begin.

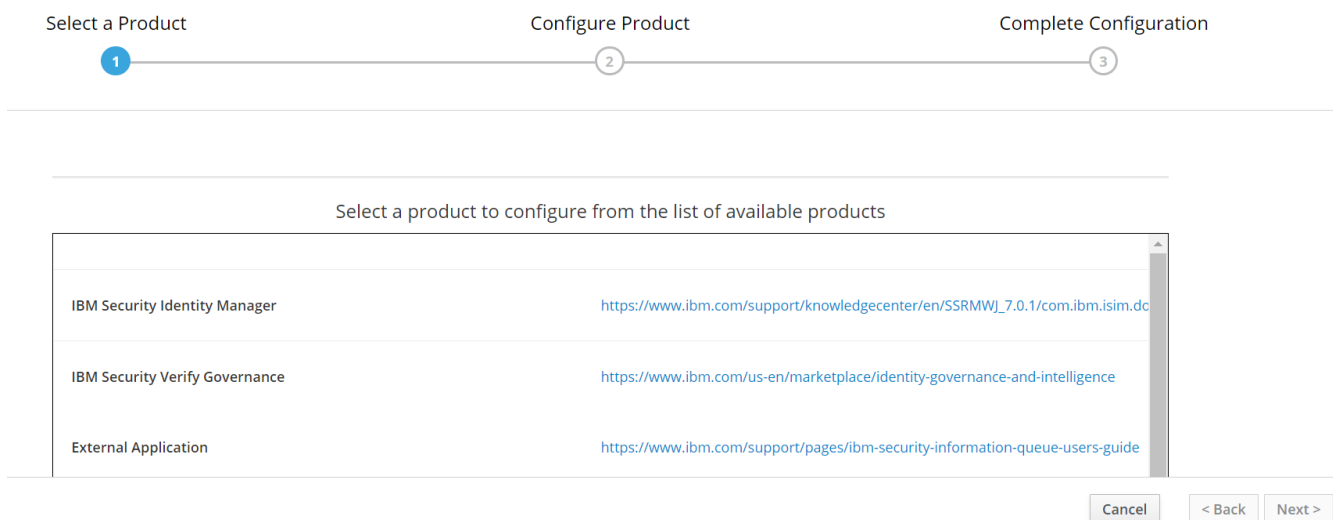
The type of login page is determined by the OIDC provider that your system administrator configured. Since I'm using ISVG's OIDC provider, the following login page is displayed:

The image shows the login page for IBM Security Verify Governance version 10.0.0.3. The page has a dark blue background with a grid pattern. On the left, the text "IBM Security Verify Governance" is displayed in white, with "10.0.0.3" below it. On the right, there is a "Log In" section with two input fields: "User Name" and "Password". Below these fields is a blue "Log In" button.

After I enter my credentials and click “Log In”, I am directed to the ISIQ UI. In this example, my ISIQ subscriber ID is “isiqAdmin”, which equals my ISVG user ID.

Select a Product

When you log in to a newly installed ISIQ with no products configured, you are redirected to the “Select a Product” page at the start of the configuration wizard:

The image shows the "Select a Product" page of the configuration wizard. At the top, there is a progress bar with three steps: "Select a Product" (marked with a blue circle and the number 1), "Configure Product" (marked with a grey circle and the number 2), and "Complete Configuration" (marked with a grey circle and the number 3). Below the progress bar, the text "Select a product to configure from the list of available products" is displayed. A table lists the available products: "IBM Security Identity Manager" with a link to the support page, "IBM Security Verify Governance" with a link to the marketplace page, and "External Application" with a link to the user guide. At the bottom right, there are three buttons: "Cancel", "< Back", and "Next >".

The available products include IBM Security Identity Manager (ISIM) and IBM Security Verify Governance (ISVG). ISIQ supports the Identity Manager (IM) component that's part of ISVG v10.0.0 or higher. For backward compatibility, ISIQ also supports ISIM v6.0 FP18 or higher, and ISIM v7.0 FP7 or higher.

ISIQ supports ISVG v10.0.0 or higher. For backward compatibility, ISIQ also supports IGI v5.2.6 FP2 or higher. ISIQ supports all database types in both IM/ISIM and ISVG/IGI.

ISIQ also offers:

1. External application support (described later in the [External Applications](#) section), which allows a client script or program to be configured as a product that consumes data or publishes data via ISIQ-supplied REST APIs.
2. An experimental Cloud Identity (CI) interface that can be leveraged to load ISIM persons as IBM Security Verify SaaS users (see [Appendix G: ISIM Integration with Cloud Identity](#)).

As ISIQ evolves, the list of available products will grow to enable you to perform additional cross-product integration.

Configuring ISIM

To use ISIQ, you must configure products. As an example, suppose you want to configure your ISIM system as a source of identity and access management (IAM) data. To do so, you highlight “IBM Security Identity Manager” on the “Select a Product” page, and click the “Next” button:

Select a product to configure from the list of available products

IBM Security Identity Manager	https://www.ibm.com/support/knowledgecenter/en/SSRMWJ_7.0.1/com.ibm.isim.doc
IBM Security Verify Governance	https://www.ibm.com/us-en/marketplace/identity-governance-and-intelligence
External Application	https://www.ibm.com/support/pages/ibm-security-information-queue-users-guide
IBM Cloud Identity (experimental)	https://www.ibm.com/us-en/marketplace/cloud-identity

Cancel < Back Next >

You then see the “Configure IBM Security Identity Manager” form. It is the most detailed of all ISIQ configuration forms since there are three separate data entities to specify:

- (1) the ISIM directory, which requires the fully qualified URL to access LDAP, an LDAP User DN with sufficient authority to retrieve the data you’re interested in, and the password for that User DN.
- (2) the ISIM web services interface, which is needed to store events that are passed from IGI. The WebServices User that you specify must be an existing ISIM user such as “itim manager” who has admin-level authority to modify accounts and persons.
- (3) the ISIM database (DB2 or Oracle), which holds audit event and process workflow information not contained in the LDAP directory.

Configure IBM Security Identity Manager

Configuration Name isim123	Description (optional) ISIM 6.0 System	Database Type ☒ DB2 ☐ Oracle
Directory Source URL ldap://isim6.ibm.com:389	WebServices URL http://isim6.ibm.com:9080	Database URL jdbc:db2://isim6.ibm.com:50000/itimdb
Directory User DN cn=root	WebServices User itim manager	Database User itimuser
Directory Password *****	WebServices Password *****	Database Password *****
Tenant DN 		Database Secure Connection ☐ Yes ☒ No

Usage Notes:

- Each ISIQ product you configure must be given a unique name, which can be up to 50 characters long and include alphanumerics as well as dashes and underscores. The Configuration Name you assign ("isim123" in this example) becomes part of any topic names created for the product.
- In this screen capture, the port numbers are specified in the URL fields. If you omit the LDAP or ISIM DB port numbers, the configuration form automatically appends a default of 389 for LDAP, and 50000/itimdb for DB2 or 1521:itimdb for Oracle. If your LDAP or ISIM DB uses a different port, then you must enter that port number here. The same is true of ISIM database names. If you use something other than "itimdb", specify that name at the end of the "Database URL".
- "isim123" is a 6.0 system. If it were 7.0, which requires SSL, you must either load an ISIM SSL certificate into the ISIQ truststore or enable the ISIQ_AUTOMATICALLY_IMPORT_CERTIFICATE environment variable in ISIQ's connect-stack.yml file. For more information, see "Secure Communication to Applications" in the [ISIQ Deployment Guide](#).
- By default, the "Database User" is assumed to be the same as the ISIM database schema name. If the two are not the same, you can override the schema by appending a ":currentSchema=" parameter to the "Database URL". For example, for an ISIM DB2 database, it would have the format: `jdbc:db2://myserver:myport/mydb:currentSchema=myschema` – for an ISIM Oracle database, you would also append `:currentSchema=myschema` at the end of the URL.
- When you configure an ISIM Oracle database, be aware that ISIQ only supports the Oracle JDBC thin driver using the colon-delimited SID format. The Service Name format is not supported.
- The "Directory User DN" must have admin-level authority to perform a sorted LDAP search based on modifytimestamp. If you are not using an admin account such as cn=root, you can still grant the required permission by setting `ibm-slapdSortSrchAllowNonAdmin` to TRUE in the `ibmslapd.conf` file on your Security Directory Server system.
- The Tenant DN field needs to be entered only if you have multiple tenants in your LDAP directory and you want to configure a particular tenant. Obtaining the Tenant DN value requires access to your ISIM system, specifically, to the `enRole.properties` file located in the `ISIM_HOME/data` directory. The Tenant DN is a concatenation of the `enrole.defaulttenant.id` and `enrole.ldapserver.root` properties. For example, if `enrole.defaulttenant.id=org` and

enrole.ldapservers.root=dc=com, the Tenant DN value is "ou=org,dc=com". If you do not have multiple tenants, leave this field blank.

After you click the “Configure” button, wait several seconds for isim123 to be added to the list of products, and for its internal tasks to start.

Connector	Status
directory	10 Task(s) Running 0 Failed
ISIMSink	10 Task(s) Running 0 Failed
ISIM_database	10 Task(s) Running 0 Failed

The “Configuration created successfully” page lists the three connectors that are started for each IBM Security Identity Manager product:

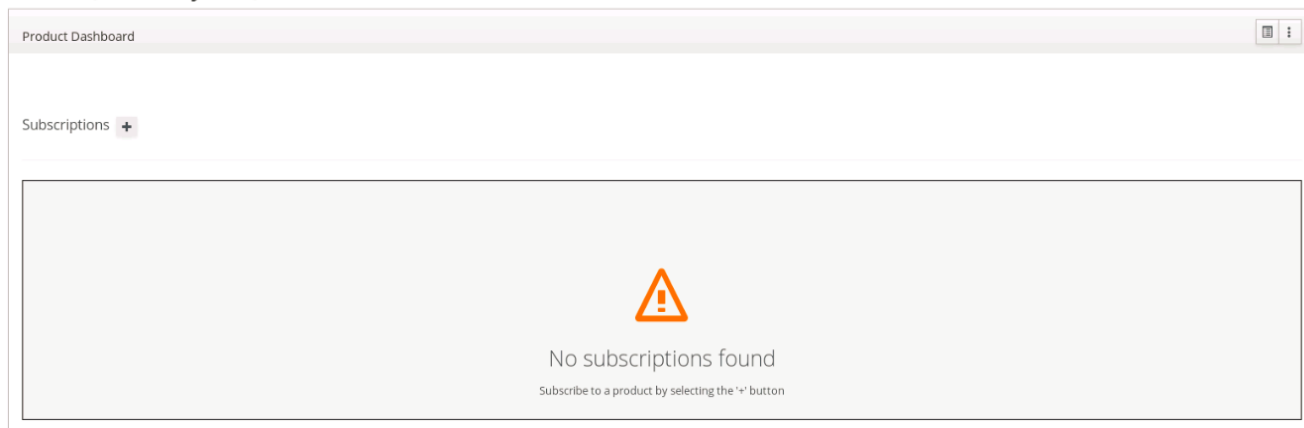
1. “directory”, which uses the directory information you configured, and acts as a data producer;
2. “ISIMSink”, which uses the web services information you configured, and acts as a consumer of events from products such as IGI;
3. “ISIM_database”, which uses the database information you configured, and acts as a data producer.

If you click the “View Product” button on this page, it takes you to the product dashboard for isim123. Every ISIQ product has its own dashboard page from which you can:

- Edit the product’s configuration information,
- Subscribe to other configured products,
- Pause a product’s tasks,
- Resume a product’s tasks,
- Pause a product’s connectors,
- Resume a product’s connectors,
- Delete a product,
- Pause a subscription,
- Resume a subscription,
- Update a subscription,
- Replay a subscription,
- Delete a subscription.

At this point, here’s what the “Subscriptions” half of the isim123 dashboard page looks like:

isim123 (ISIM 6.0 System)

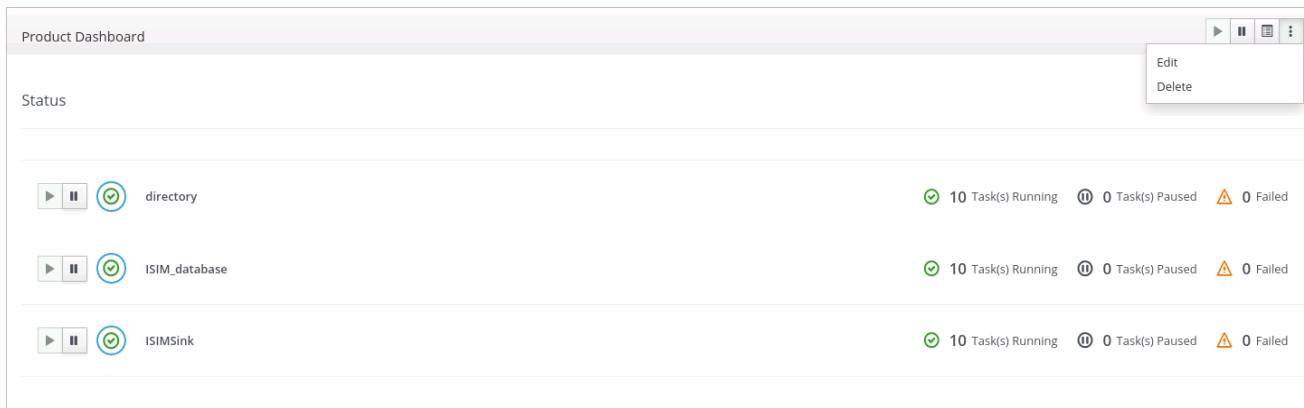


Since only the one IBM Security Identity Manager product is configured, clicking the ‘+’ button next to “Subscriptions” shows an empty drop-down list because there’s nothing yet to subscribe to. You must configure at least one other product to set up subscriptions.

Clicking the icon in the upper right of this page displays a menu with “Edit” and “Delete” under “Product Options”:

isim123 (ISIM 6.0 System)

Owner: isiqAdmin



Suppose you discover you entered the wrong password for the Directory User DN when you configured isim123, and you now want to correct it. In that case, select the “Edit” menu option, which displays all of your isim123 configuration settings:

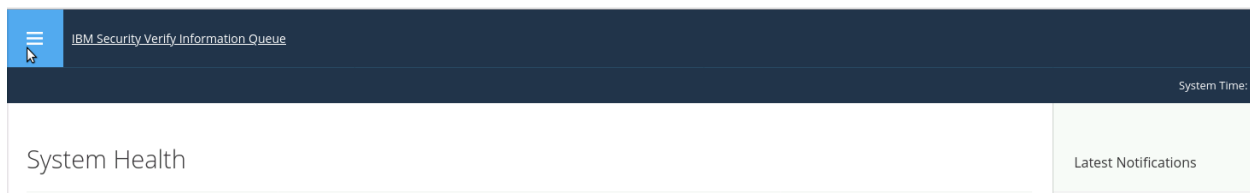
Configuration Name isim123	Description (optional) ISIM 6.0 System	Database Type ☒ DB2 ☐ Oracle
Directory Source URL ldap://isim6.ibm.com:389	WebServices URL http://isim6.ibm.com:9080	Database URL jdbc:db2://isim6.ibm.com:50000/itimdb
Directory User DN cn=root	WebServices User itim manager	Database User itimuser
Directory Password ••••••••	WebServices Password •••••	Database Password ••••••••
Tenant DN 	Database Secure Connection ☐ Yes ☒ No	

Type the correct password and click the “Save” button. The change immediately goes into effect. (Note: When you invoke the "Edit" page for any ISIQ product, you must always reenter all password credentials. The page does not populate previously saved passwords.)

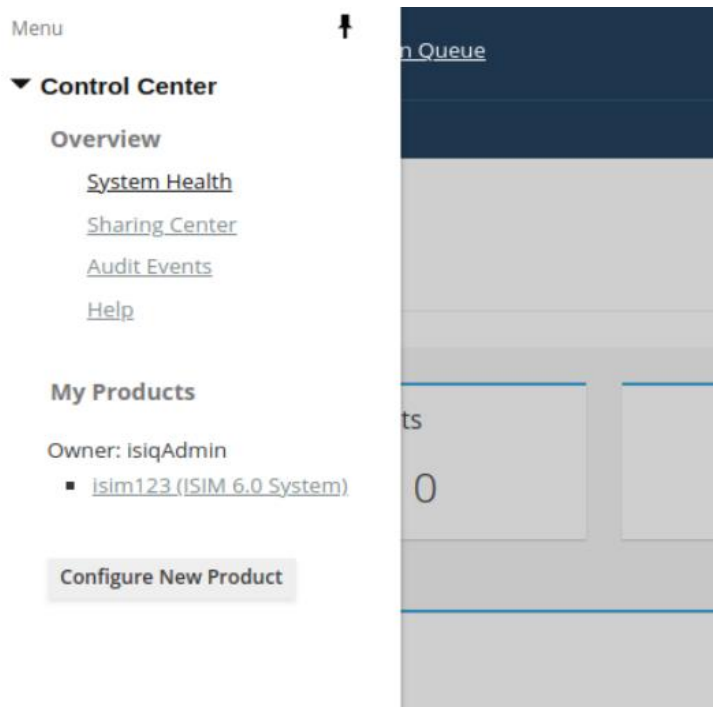
Configuration Name is a key value in ISIQ and is not editable. To specify a different Configuration Name, you must first select the “Delete” menu option and then reconfigure the IBM Security Identity Manager product.

Note on ISIQ Navigation

From any page in the ISIQ user interface, you can click the “hamburger” icon in the upper left (highlighted in blue in this screen capture):



You then see a navigation sidebar as shown here:



The sidebar includes several useful links, such as “System Health” to direct you to the ISIQ System Health dashboard, “Sharing Center” to manage product sharing among ISIQ user groups, “Audit Events” to browse key actions in the life of ISIQ (for more information, see “ISIQ Event Auditing” in the [ISIQ Deployment Guide](#)), and a “Configure New Product” button that returns you to the “Select a Product” page. To hide the navigation sidebar, click anywhere on the main page to the right of the sidebar.

Verifying ISIM Data Loading

Before proceeding further with ISIM-to-IGI integration, we recommend you pause at this point and verify that ISIQ is loading all expected IBM Security Identity Manager data from your LDAP directory. A number of issues can arise when ISIQ queries LDAP, and it is better to find and fix those issues early. For example:

- There might be mismatches between your schema and the LDAP directory. The mismatches can cause object profiles to not be read by ISIQ. Sometimes there are old schema definitions, which are ignored by ISIM but that interfere with ISIQ’s ability to process all of your profiles. ISIQ implementations offer an opportunity to clean up schema discrepancies of this sort.
- You might have a custom attribute that doesn’t have a dotted number OID notation, nor does its name end with an “-oid” suffix. Consequently, ISIQ can’t retrieve ISIM entries with that attribute because of LDAP parsing errors. The remedy is to stop LDAP, edit the V3.modifiedschema file so that the attribute definition has a name ending with “-oid”, and then restart LDAP.
- There might be resource constraints on the Security Directory Server (SDS) system. For instance, if ISIQ’s LDAP query finds hundreds of thousands of entries for a particular profile, those entries must first be sorted in sequence by modifytimestamp. A shortage of sort heap memory delays completion of the sort and the query. Also, if the modifytimestamp index wasn’t created during ISIQ deployment (see “Tuning Requirements” in the [ISIQ Deployment Guide](#)), that too can cause

the default 30 second timeout for an ISIQ LDAP query to be exceeded. The remedy is to add memory to the SDS system if necessary, and to create the modifytimestamp index to optimize ISIQ's LDAP searches.

- The number of available LDAP connections might not be sufficient. To accelerate data integration, ISIQ starts ten threads to run LDAP queries in parallel. If your SDS does not have the extra connection capacity, some LDAP queries from ISIQ will hang. You will see a growth in open TCP connections when you run `netstat` commands on the SDS system and grep for the LDAP listening port. The remedy is to configure more connections in SDS.

To help you check that ISIM data loading is working properly, there are two shell scripts, `topicList.sh` and `topicPoll.sh`, located in `<starterKitDir>/util`:

(1) Soon after you configure an IBM Security Identity Manager product in ISIQ, run the ``topicList.sh`` command. You should see a list of topics created with "`<isimProductName>.directory`" included in the full topic name (for more information about topics, see "Connectors and Topics" later in this guide). ISIQ creates and loads a topic for each non-empty object profile found in ISIM's LDAP.

Compare the "directory" topics in the ``topicList.sh`` output against your knowledge of the profiles in LDAP. Are there expected topics missing from the `topicList.sh` output? For example, if you know that you have many AD accounts in ISIM, but you don't see an `ADAccount` topic, it suggests that ISIQ is unable to complete its LDAP search for AD accounts.

(2) If you see an `ADAccount` topic, does the topic have the right number of entries in it? To answer that question, copy the full `ADAccount` topic name and paste it into a ``topicPoll.sh`` command: ``topicPoll.sh <ADAccountTopicName>`` (if the command will take longer than the default 15 seconds to retrieve all entries, add the `-f` flag to the command line, which lets the script run until you use Ctrl-C to terminate it).

Note: As a best practice when running `topicPoll.sh`, we recommend you copy-and-paste the full topic name from the `topicList.sh` output. This guarantees that the name will be entered correctly. If there's a typographical error in the topic name that you pass to `topicPoll.sh`, even something as minor as an uppercase/lowercase change, Kafka will create an empty topic with the incorrect name. If this problem arises, you can remove the empty topic by running the `<starterKitDir>/util/topicDelete.sh` script.

If the `topicPoll.sh` output contains too many entries to scan through, pipe the output to a file and run ``wc -l`` or a similar utility to get a count of entries in the file. Does the count match the number reported in the IBM Security Identity Manager UI? If not, there might be an issue, as described previously, in which your LDAP has OID attributes without an "-oid" suffix in their names. As a result, ISIQ's LDAP query terminates with an error and doesn't return all entries.

The advantage of verifying IBM Security Identity Manager data loading at this point, before you configure IGI and before you define a subscription, is that

- (a) You don't run the risk of sending incorrect or incomplete IBM Security Identity Manager data to your IGI that will require cleanup actions later.
- (b) It's easier to analyze ISIQ's connect service log since it includes tracing for only the one IBM Security Identity Manager product. It can be helpful to temporarily set `CONNECT_LOG4J_ROOT_LOGLEVEL=DEBUG` instead of running with the default INFO logging level, so that you have more useful diagnostic data in the log (see the "connect-stack.yml" section in Appendix A: YAML Files" of the [ISIQ Deployment Guide](#)). Once you verify that ISIQ is collecting all of the expected LDAP data, reset the logging level back to INFO.

Configuring IGI

Continuing the ISIQ setup, after you verify that all isim123 data is being loaded properly into topics, you now want to configure an IGI product so that it can subscribe to isim123 and perform cross-product integration. After you click “Configure New Product” from the navigation sidebar, highlight “IBM Security Verify Governance” on the “Select a Product” page:

Select a product to configure from the list of available products

IBM Security Identity Manager	https://www.ibm.com/support/knowledgecenter/en/SSRMWJ_7.0.1/com.ibm.isim.doc
IBM Security Verify Governance	https://www.ibm.com/us-en/marketplace/identity-governance-and-intelligence
External Application	https://www.ibm.com/support/pages/ibm-security-information-queue-users-guide
IBM Cloud Identity (experimental)	https://www.ibm.com/us-en/marketplace/cloud-identity

Cancel < Back Next >

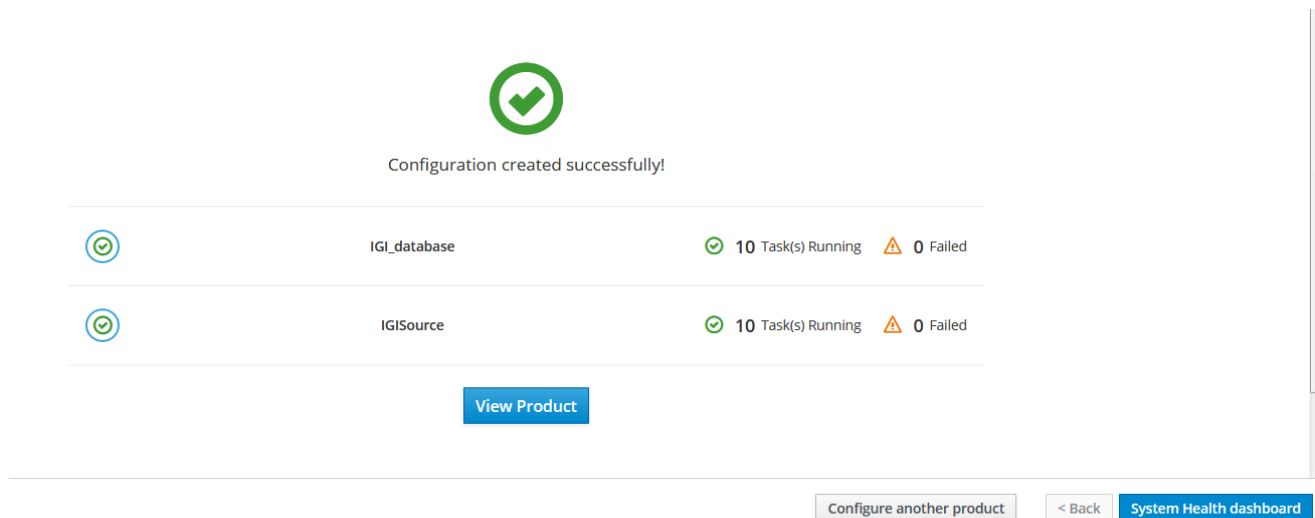
After you click the “Next” button, you must fill in the IGI configuration form:

Configure IBM Security Verify Governance

Configuration Name	Database URL
IGI_Test	jdbc:db2://testigi526.ibm.com
Description (optional)	Database User
description of product being configured	igacore
Database Type	Database Password
☒ DB2 ☐ Oracle ☐ PostgreSQL	
	Database Secure Connection
	☐ Yes ☒ No

Cancel < Back Configure >

Most of the form’s requested information pertains to the IGI database (DB2, Oracle, or PostgreSQL) because that is where the data produced and consumed by IGI is stored. After you click “Configure”, wait several seconds for IGI_Test to be added to the list of products, and for its internal tasks to start.



Configuration created successfully!

	IGI_database	 10 Task(s) Running  0 Failed
	IGISource	 10 Task(s) Running  0 Failed

[View Product](#)

[Configure another product](#) [< Back](#) [System Health dashboard](#)

The “Configuration created successfully” page lists the two connectors that are started for each IGI product:

1. “IGI_database”, which uses the database information you configured, and acts as a consumer of data from products such as IBM Security Identity Manager;
2. “IGISource”, which also uses the configured database information, but acts as a producer of data for products that subscribe to IGI events.

By default, the IGISource connector reads from the USER_EVENT_ERC table to collect Account events. In the ISIQ connect-stack.yml file, you can optionally enable the connector to also read from the EVENT_OUT_USER table to collect User events. If you configure only one-way ISIM-to-IGI integration, the IGISource connector still collects and publishes events, but ISIM does not consume the event data.

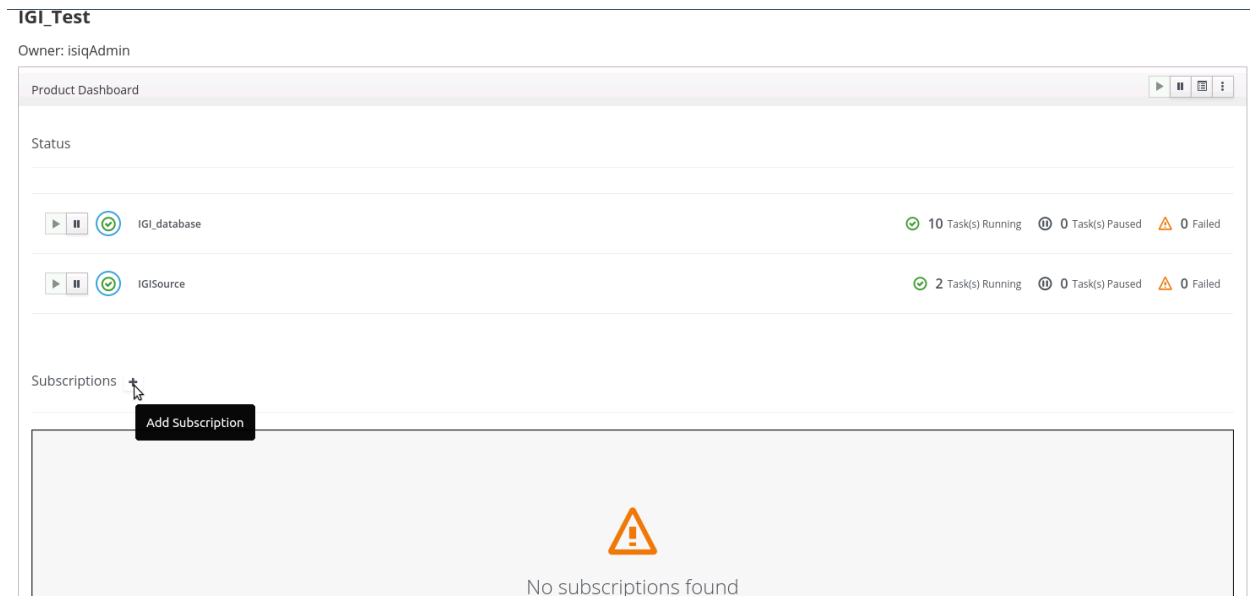
Usage Notes:

- Each ISIQ product you configure must be given a unique name, which can be up to 50 characters long and include alphanumerics as well as dashes and underscores. The Configuration Name you assign (“IGI_Test” in this example) becomes part of any topic names created for the product.
- The out-of-the-box IGI product must be customized in several ways to interoperate with ISIQ. For a list of required customizations, see “Appendix A: IGI Customizations for ISIQ”.
- When you configure an IGI Oracle database, be aware that ISIQ only supports the Oracle JDBC thin driver using the colon-delimited SID format. The Service Name format is not supported.

Subscriptions

Now that you configured a second product, you can define a subscription. Subscriptions are at the heart of what ISIQ is about, and where the real value emerges in cross-product integration. If you configure products without defining subscription relationships between them, the products will publish data but there will be no consumers of the data. Subscriptions also provide a security mechanism to ensure only trusted consumers, which you configured, are allowed to access published data.

To define a subscription, click the “View Product” button to take you to the IGI_Test product dashboard (note: you can also access the IGI_Test dashboard by clicking the “IGI_Test” link under “My Products” in the navigation sidebar).



When you click the ‘+’ next to “Subscriptions” (it has “Add Subscription” as its help text), your isim123 product appears in the drop-down list as an eligible source to subscribe to. Select the product, which enables the “Subscribe” button in the dialog:

Add a subscription for IGI_Test

- ☐ Subscribe directly to the data source (no transformation)
- ☒ Subscribe to the transformed data source (default transformation)
- ☐ Subscribe to the custom transformed data source (custom transformation)

Select a source to subscribe to:

Subscribe Cancel

Note: “default transformation” is currently the only supported subscription type.

IBM Security Identity Manager source names have the format, <subscriberID>.<configurationName>.ISIM. After you click “Subscribe”, ISIQ returns you to the IGI_Test product dashboard with the new subscription listed, as shown here:

IGI_Test

Owner: isiqAdmin

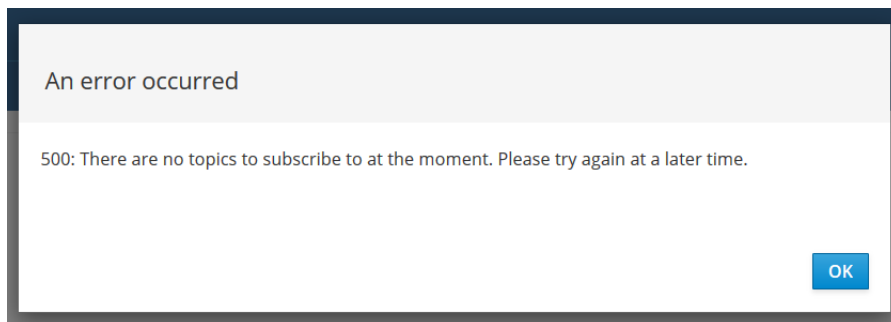
The screenshot shows the 'Product Dashboard' for 'IGI_Test'. At the top, it says 'Owner: isiqAdmin'. Below the title bar, there's a 'Status' section. It lists two components: 'IGISource' and 'IGI_database'. For 'IGISource', it shows '2 Task(s) Running', '0 Task(s) Paused', and '0 Failed'. For 'IGI_database', it shows '10 Task(s) Running', '0 Task(s) Paused', and '0 Failed'. Below the status section is a 'Subscriptions' section with a '+' icon. At the bottom, there's a table with two columns: 'isim123' and 'isim'. The 'isim123' column has a 'Reprocess' button and a play button. The 'isim' column has a play button. The last updated time is '2021-02-15 16:50 GMT'.

You now have cross-product integration.

You can also integrate in the reverse direction, namely, go to the isim123 product dashboard and select IGI_Test from the Subscriptions drop-down. That makes IGI the producer and ISIM the subscribing consumer. IGI_Test would send its account and entitlement-related events to the ISIM web services API that you configured for isim123. For more information, see [Appendix C: Mapping Entities and Events between ISIM and IGI](#).

Usage Notes:

(1) If you have configured a product that does not yet have source topics (see the subsequent “Connectors and Topics” section for a discussion of Kafka topics), any attempts to “Subscribe” to that product will fail with the following message box displayed:



This scenario most commonly happens when you attempt to subscribe ISIM to IGI at the same time as you subscribe IGI to ISIM. IGI has two possible source topics: EVENT_OUT_USER (for user events) and USER_EVENT_ERC (for account events). Those topics don’t exist until the ISIM-to-IGI subscription runs for a while and generates one or both of the event types in IGI (see the Appendix A section, “Enable the EVENT_OUT_USER table”, for instructions regarding the enablement of user events). Once ISIM pushes data to IGI, or you manually create an IGI user or change an IGI account, and you start seeing entries appear in the Access Governance Core -> Monitor -> OUT events tab, then you can subscribe ISIM to IGI.

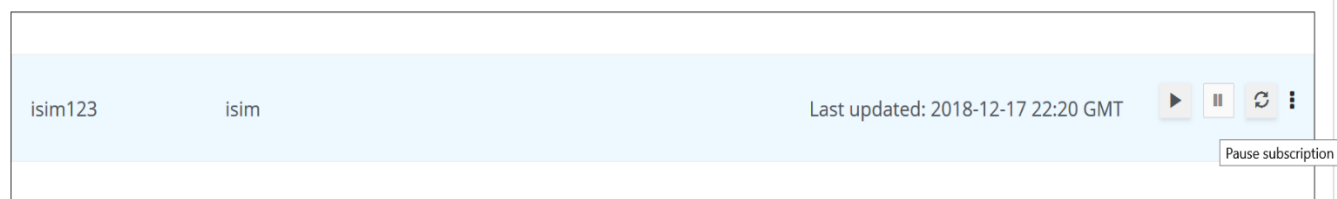
This timing issue doesn't occur when you subscribe IGI to ISIM because ISIM's source topics are created as soon as the product is configured.

(2) If you subscribe IGI to ISIM, and then later import more ISIM adapters, IGI won't automatically have access to topics published by those adapters. To remedy the situation, click the "Update subscription" button on the IGI product dashboard page (refer to [Appendix D: Updating Subscriptions](#)). That causes a re-read of all the latest available ISIM topics.

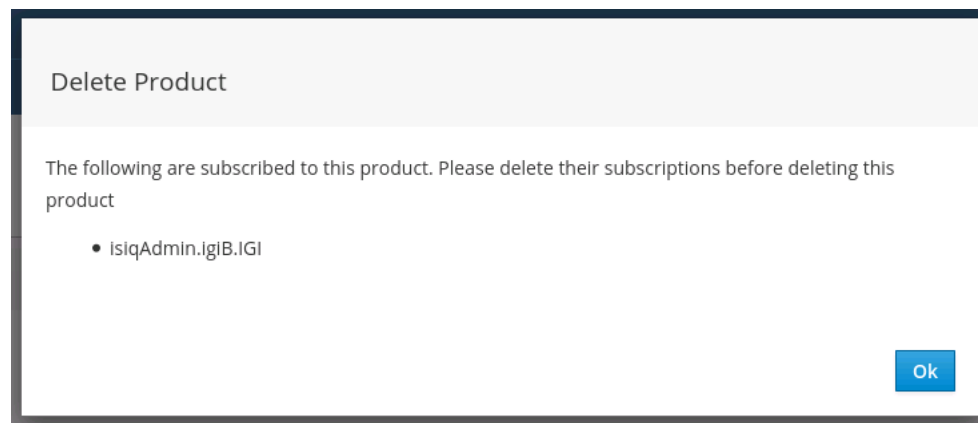
(3) When you define a subscription, it is active by default. If necessary, you can later pause and resume it. Pausing a subscription temporarily stops all data integration from the producer to the consumer. You might want to pause a subscription if, for instance, your producer/source is being reset or is undergoing maintenance and must not publish data during that period. You can then resume the subscription once the maintenance window completes.

After you click "Pause subscription" or "Resume subscription", a message box confirms that the operation was executed. You can determine the current state of a subscription by the appearance of the buttons themselves. If a subscription is active, the "Pause subscription" button is enabled, and the "Resume subscription" button is disabled. If a subscription is paused, the button states are reversed, as shown in this screen capture where the "Resume subscription" button  is enabled/clickable, and the "Pause subscription" button is disabled:

Subscriptions 



(4) If a product is being subscribed to, it cannot be deleted because doing so disrupts subscribers. For example, if igiB is subscribed to isimC, attempting to delete isimC displays a message box like this:



All subscriptions to a product must be deleted before the product can be deleted.

(5) When you subscribe an IGI product to ISIM, the ISIM topic data starts being inserted into IGI. If you're in test mode where you expect to perform multiple ISIM-to-IGI data loads, it's a best practice, before you define the subscription, to first take a snapshot or backup of your IGI data tier. Doing so will let you quickly restore the initialized IGI environment before running the next ISIM-to-IGI data load.

Connectors and Topics

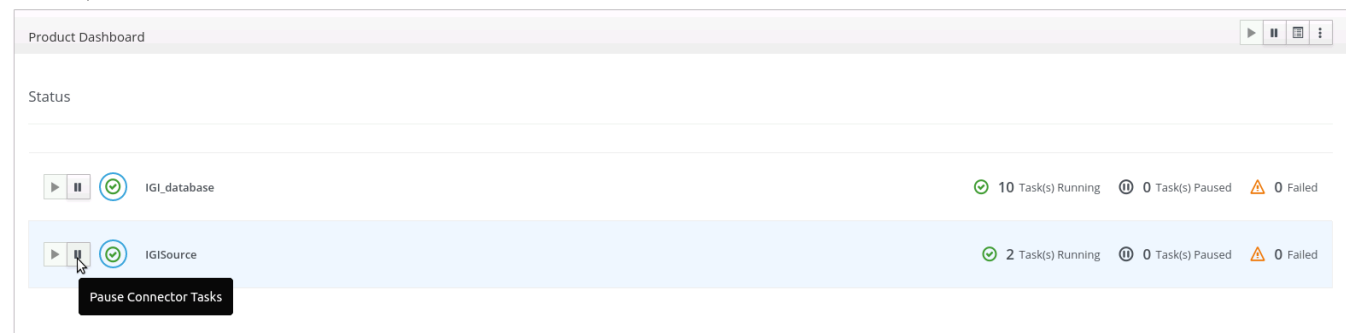
In ISIQ, when you configure an ISIM or IGI product, you are in fact configuring one or more ISIQ-supplied connector programs. In Kafka terminology, there are source connectors and sink connectors. Source connectors produce and publish data. Sink connectors subscribe to and consume data. For each ISIM product you configure, ISIQ includes source connectors that take data from LDAP and from the ISIM database and push it into Kafka “topics” (collections of related data). ISIQ also includes a sink connector that pushes data to ISIM using its web services API. For each IGI product you configure, ISIQ includes a sink connector that pulls data from topics and stores it in the IGI database, and a source connector that forwards IGI events to topics so that a subscribing product like ISIM can consume them.

On the dashboard page for a product that has connectors, you will notice there are two buttons, “Pause Connector Tasks” and “Resume Connector Tasks”, next to each connector. Although these buttons are not commonly required, there might be scenarios where they are helpful. For example, suppose you have two-way integration between ISIM and IGI, and you want to temporarily stop the IGI data source so that you can remove certain IGI OUT events before they get sent to ISIM. In that case, click the “Pause Connector Tasks” button next to “IGISource”:

IBM Security Verify Governance

IGI_Test

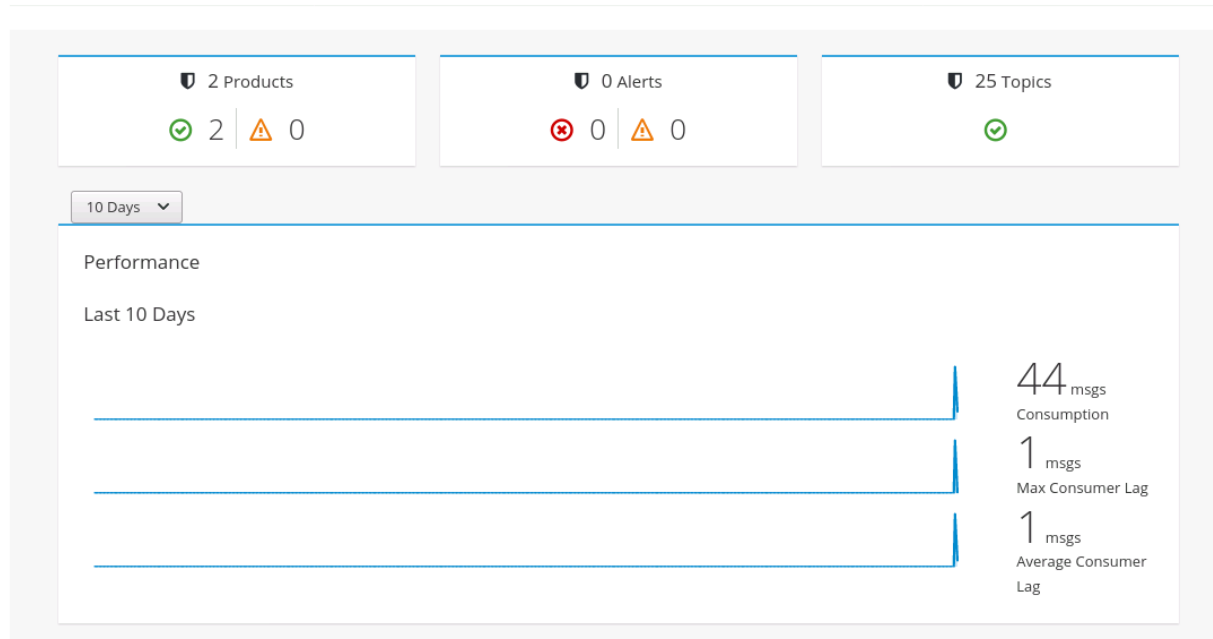
Owner: isiqAdmin



While the connector is paused, the only data flow is from ISIM to the IGI database. After you remove the IGI OUT events, click the “Resume Connector Tasks” button to restart two-way integration.

Continuing our example, as a result of configuring isim123 and IGI_Test, there are now 25 topics (note: the number of topics varies depending on the number of adapter profiles imported into your ISIM). To see what they are, you can select the “25 Topics” link on the System Health dashboard:

System Health



A scrollable list appears:

Topic List			
IsiqAdmin.Isim123.isim_database.eventCompletion	N/A	N/A	N/A
IsiqAdmin.Isim123.directory.CSVFeed	1	0	0
IsiqAdmin.Isim123.directory.DefaultRole	4	0	0
IsiqAdmin.Isim123.directory.ITIMAccount	35	0	0
IsiqAdmin.Isim123.directory.ITIMGroup	5	0	0
IsiqAdmin.Isim123.directory.ITIMProfile	1	0	0
IsiqAdmin.Isim123.directory.Organization	1	0	0
IsiqAdmin.Isim123.directory.OrganizationalUnit	3	0	0
IsiqAdmin.Isim123.directory.Person	21	0	0
IsiqAdmin.Isim123.directory.PosAixGroupProfile	39	0	0
IsiqAdmin.Isim123.directory.PosAixRoleProfile	13	0	0
IsiqAdmin.Isim123.directory.PosixAixAccount	21	0	0
IsiqAdmin.Isim123.directory.PosixAixProfile	1	0	0

OK

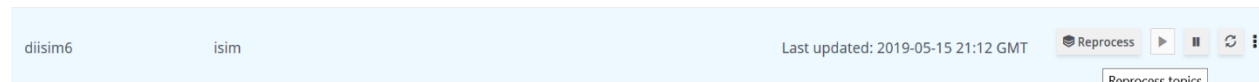
In addition to displaying topic names, the list contains “Msgs”, “Warn”, and “Fail” columns that itemize the number of messages, warnings, and failures per topic.

The naming pattern for ISIM directory topics is <subscriberID>.<configurationName>.directory.*. If you examine the last level of the names, you will observe that it includes (a) general LDAP topics such as “Organization” and “Person”, and (b) topics associated with Security Verify Adapters like the one for UNIX-Linux. If you imported into ISIM other Security Verify Adapters such as for Windows AD, SQL

Server, Google Apps, RACF, or PeopleSoft HR, then the list of generated topics automatically grows to include those other adapter profiles.

Reprocessing Topics

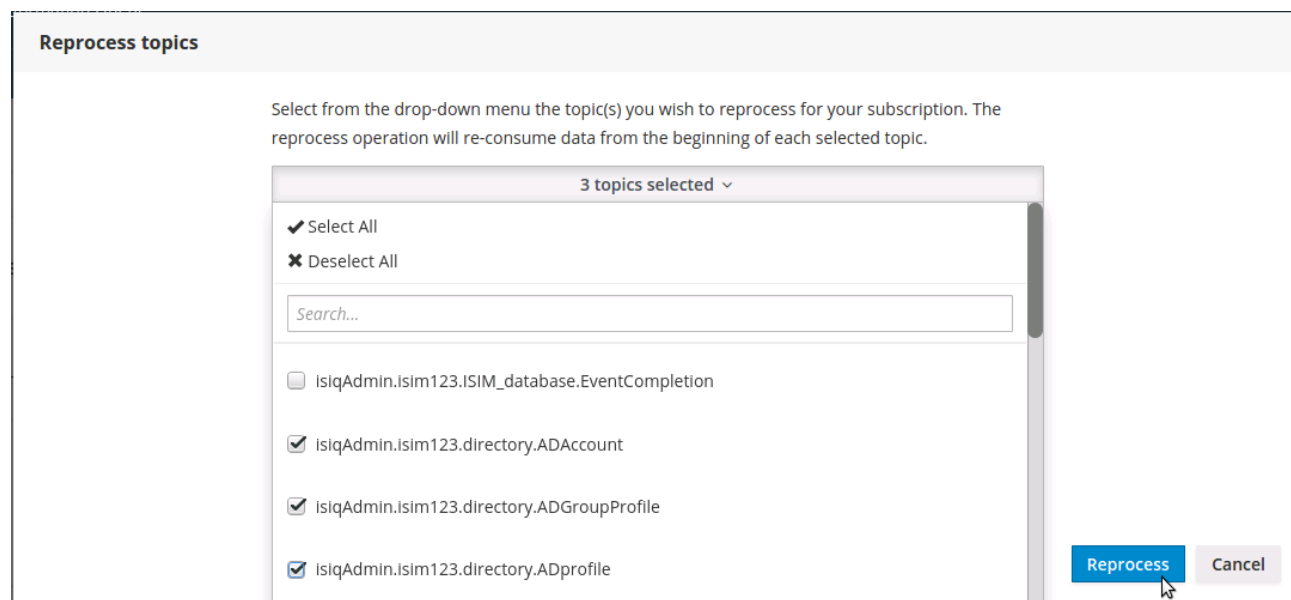
ISIQ allows subscribers to reprocess topic data from a product to which they are subscribed. On the dashboard for a connector-based product such as ISIM or IGI, there is a “Reprocess” button for any subscriptions you defined (note: reprocessing is not applicable to ISIQ external applications).



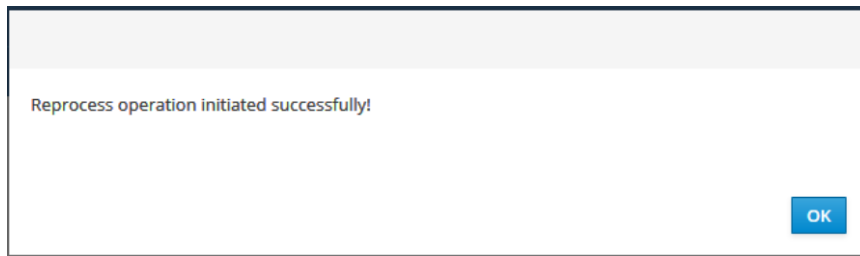
“Reprocess” is essentially a replay capability that can be beneficial in the following use-case scenarios:

- (1) You’d like to reload a select set of ISIM entities into IGI after customizing how transformations on those entities are performed (refer to the “Custom Transformations” section for details).
- (2) You’ve enabled a new attribute-to-permission mapping configuration in IGI and want to apply policy changes to data affected by the new mappings.
- (3) You’re recovering from a failure and want to be sure that your producer and consumer endpoints are synchronized.

Clicking “Reprocess” displays a dialog with a drop-down menu. After you expand the drop-down, you see a list of source topics eligible for reprocessing. Each topic has a checkbox next to it:

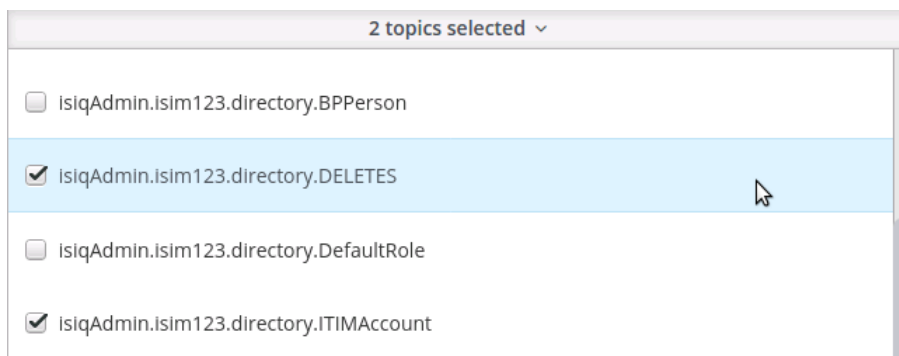


Once you’ve checked one or more topics and clicked “Reprocess”, ISIQ initiates an operation that resets each selected topic’s offset to zero, and re-consumes all of the topic’s data:



“Reprocess” takes a little while to run and triggers a rebalance in ISIQ’s connector tasks, which temporarily stops data integration. Therefore, it is recommended that you use “Reprocess” sparingly, only if necessitated by one of the previously listed use-case scenarios, and only after your initial product-to-product integration completes.

When you are reprocessing ISIM source data, be aware of the special properties of the “DELETES” topic. It contains records for all other ISIM topics that had delete activity:



If you’re reprocessing any ISIM topic (“ITIMAccount” in the screen capture), you normally want to also select “DELETES” since it might contain records applicable to the topics you’re reprocessing. For example, if an Active Directory account was created in ISIM, modified, and then later deleted, you expect that when reprocessing finishes for the ADAccount topic, the deleted AD account will still be deleted. However, if you didn’t select “DELETES”, that account will show up again in your subscriber because the create and modify account events in the ADAccount topic get replayed.

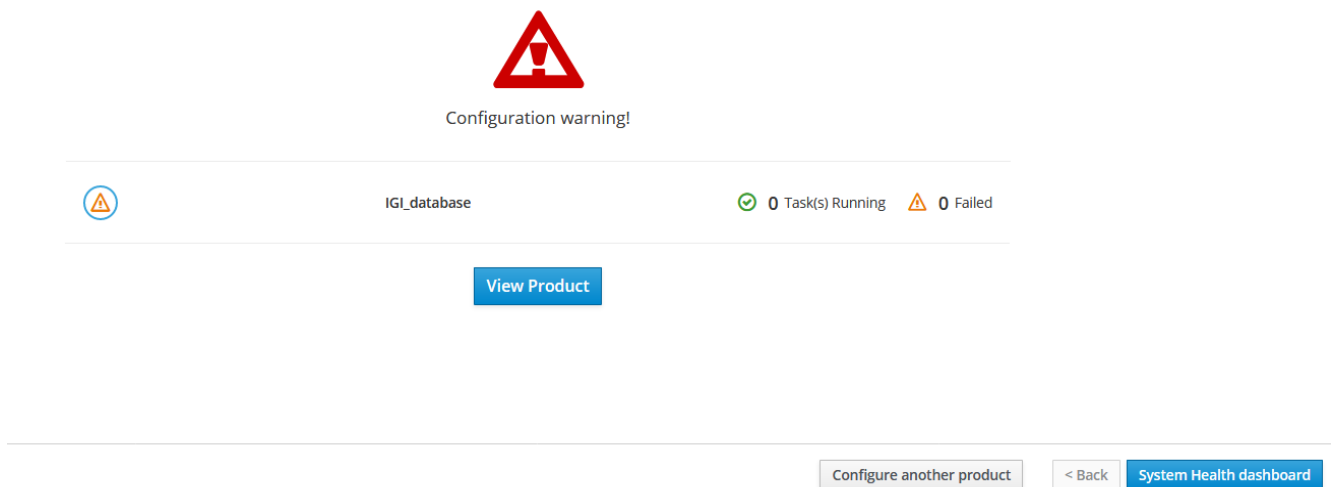
Note: There’s no harm if “DELETES” contains records for topics you did not select in the drop-down menu. Those records won’t have an effect. They will attempt to delete non-existent entities in the subscriber, which is an ignorable condition during reprocessing.

One exceptional circumstance would be if some entities on your system were deleted by mistake, and you’re reprocessing ISIM topics in order to recover those entities. In that case, do not select the “DELETES” topic since you don’t want the deletes to be replayed.

Verifying Product Registration

After you configure an ISIM or IGI product, verify that it registered correctly. Here are simple steps you can take:

(1) **Check the status page after you click the “Configure” button.** Certain problems, such as specifying the wrong database port number when you configure IGI, generate a Warning status:



The “Configuration warning!” message results from repeated failed attempts by the IGI sink connector to connect to the database. A similar warning appears on the dashboard page for this product.

(2) **Check for running tasks associated with the configured product.** In this screen capture, we see that 0 tasks are running, which signifies unsuccessful product registration.

(3) **Ensure that topics were created.** If the status page displays, “Configuration created successfully”, you can then open the “Topic List”, as shown previously, to confirm there are topics that contain the configuration name you specified. If you see one or more topics listed there, it’s a good indication that product registration worked correctly.

If product registration fails and you’re not sure why, you might need to contact your ISIQ system administrator who has the authority to issue docker commands on the ISIQ host. For more information about diagnosing product registration problems, see the [ISIQ Troubleshooting Guide](#).

Transformers

ISIQ also supplies a set of default transformers. They are a specialized type of connector generated during product subscription. ISIQ transformers perform single entity mappings to reformat or rename source topic data so that it’s more readily consumed by a sink connector or external application.

In simple data integration scenarios, you might not notice transformers. They work behind-the-scenes to take, for example, a Person topic attribute from product A, and publish it as a User topic attribute for product B.

If you run a query against `http://localhost:8083/connectors/` to list all of your ISIQ connectors, you see output like this:

```
[
  "isiqAdmin.IGI_Test.IGISource",
  "isiqAdmin.efffaaa5-4b39-44e0-b149-d79c50a61fc6.transformer",
  "isiqAdmin.isim123.directory",
  "isiqAdmin.IGI_Test.IGI_database",
  "isiqAdmin.isim123.ISIM_database",
  "isiqAdmin.isim123.ISIMSink"
]
```

Connectors with a GUID in their names are the internally generated transformers. For instance, here's what a query against `http://localhost:8083/connectors/isiqAdmin.efffaaa5-4b39-44e0-b149-d79c50a61fc6.transformer` reveals:

```
{
  "config": {
    "bootstrap.servers": "kafka1:9092",
    "connector.class": "com.ibm.kafka.connect.transformer.TransformerSink",
    "name": "isiqAdmin.efffaaa5-4b39-44e0-b149-d79c50a61fc6.transformer",
    "source.id": "isiqAdmin.isim123.directory",
    "source.type": "isim",
    "target.id": "isiqAdmin.IGI_Test.IGI_database",
    "target.type": "igi",
    "tasks.max": "10",
    "topics": "isiqAdmin.isim123.ISIM_database.EventCompletion,isiqAdmin.isim123.directory.ADAccount,isiqAdmin.isim123.directory.ADGroupProfile,isiqAdmin.isim123.directory.ADprosim123.directory.BPPerson,isiqAdmin.isim123.directory.DELETES,isiqAdmin.isim123.directory.DefaultRole,isiqAdmin.isim123.directory.ITIMAccount,isiqAdmin.isim123.directory.ITIMGroup,i,isiqAdmin.isim123.directory.ITIMProfile,isiqAdmin.isim123.directory.Location,isiqAdmin.isim123.directory.Organization,isiqAdmin.isim123.directory.OrganizationalUnit,isiqAdmin.isim123.directory.Perso123.directory.RACFacct,isiqAdmin.isim123.directory.RACFgroupProfile,isiqAdmin.isim123.directory.racf2profile"
  },
  "name": "isiqAdmin.efffaaa5-4b39-44e0-b149-d79c50a61fc6.transformer",
  "tasks": [
    {
      "connector": "isiqAdmin.efffaaa5-4b39-44e0-b149-d79c50a61fc6.transformer",
      "task": 0
    },
    {
      "connector": "isiqAdmin.efffaaa5-4b39-44e0-b149-d79c50a61fc6.transformer",
      "task": 1
    }
  ]
}
```

This screen capture shows that the transformer was generated for `source.type "isim"` and `target.type "igi"`, which correspond to "in" and "out" keys in the ISIQ-supplied `txdef.json` file. The file includes template definitions with default ISIM-to-IGI mapping rules for all ISIM-compatible Security Verify Adapters. For more information about `txdef.json` and how it maps attributes between products, see the [Custom Transformations](#) section. For specifics on how ISIM data gets loaded into IGI, see [Appendix C: Mapping Entities and Events between ISIM and IGI](#).

ISIM-to-IGI Integration

Using our example, let's recap how the ISIQ end-to-end flow operates to implement ISIM-to-IGI integration:

1. After you configured `isim123`, an LDAP source connector and a database source connector began running and publishing data into a set of source topics.
2. After you configured `IGI_Test`, an IGI database sink connector began running.
3. After you subscribed `IGI_Test` to `isim123`, the ISIM source topics started being consumed by the IGI sink connector.
4. The ISIM source topics were also transformed, by using the ISIQ-supplied default-mapping rules, into a format compatible with IGI.
5. Any subsequent changes to `isim123` (for example, creating a new user account) are read by the IGI sink connector and inserted into the `IGI_Test` database.
6. If you also subscribe `isim123` to `IGI_Test`, you have two-way integration in which IGI acts as a source to forward its account and entitlement-related events to ISIM.

As a result of configuring ISIM and IGI products, and setting up subscription relationships between them, you are able to integrate ISIM and IGI in a manner analogous to the IBM Security Identity Governance and Administration Data Integrator (ISIGADI).

External Applications

ISIQ also supports external applications. An external application doesn't include an ISIQ-supplied connector. Instead, the application itself is the Kafka consumer, and you subscribe it to a product such as IBM Security Identity Manager that has source topics. External applications communicate with ISIQ via REST APIs. The main API to use has this format:

<https://<HostName>/api/external/consume/<TopicName>?<optionalQueryParameters>>

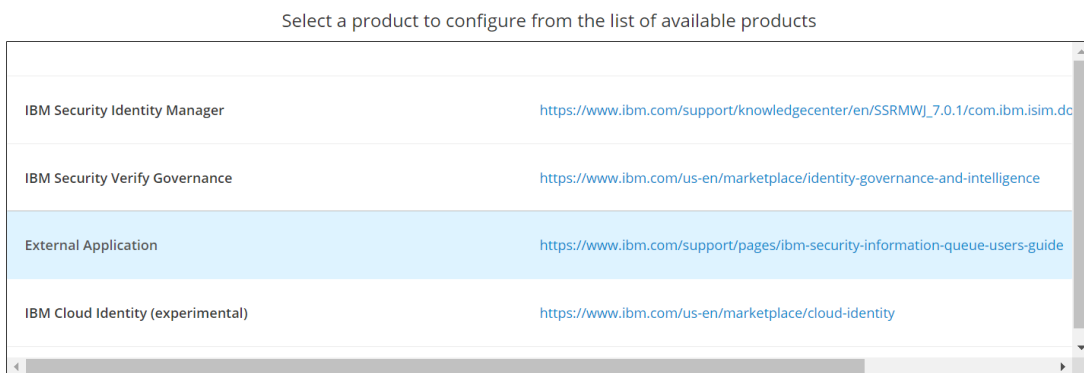
You can decide which topics to consume by viewing the “Topics” link on the System Health dashboard, as shown earlier in the “Connectors and Topics” section.

Alternatively, if your application needs to publish data to a topic, then your API calls use this format:

<https://<HostName>/api/external/produce/<TopicName>>

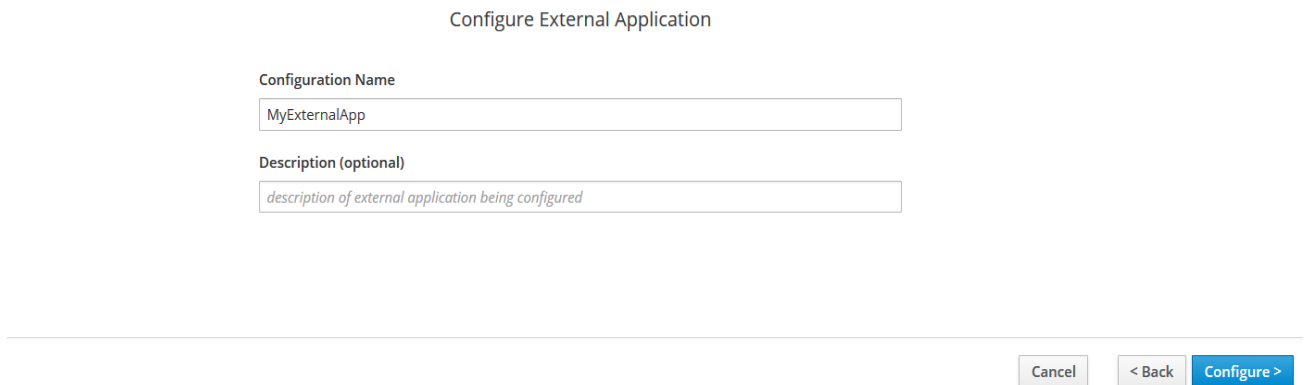
To ensure secure access to ISIQ's REST API, here are the steps to follow:

1) Configure your application as an ISIQ product by choosing “External Application” from the “Select a Product” page:



Cancel < Back Next >

After you click the “Next” button, you see how the configuration setup for external applications is minimal. All that's required is a unique Configuration Name, and then click the “Configure” button:



The diagram illustrates a three-step configuration process:

- Select a Product**: Step 1, indicated by a blue circle with the number 1.
- Configure Product**: Step 2, indicated by a blue circle with the number 2.
- Complete Configuration**: Step 3, indicated by a blue circle with the number 3.

A large green checkmark icon is centered below the steps, signifying successful completion. Below the icon, the text reads: "Configuration created successfully!"

Please use the following API token when producing and consuming messages to ISIQ:

`eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQIOjkiYW5iaXNrcyUiTXlFeHRlcm5hbnEwC5FWFRBUFAILCjpc3MiOjpc2xiwlaWF0IjozMNTM1NTc3NDgyYQ.-9M_xrleisj5ZbvhtEPbdS_kCep7D_JHDyoDDwLm3dA`

The client is responsible for storing and keeping the API token in a safe place

If needed, you may generate a new API token from the product dashboard

A button labeled "View Product" is located at the bottom center of the diagram area.

Note: If you misplace the token, or want to generate a new one, you can do so on the product dashboard page by clicking “Regenerate API Token”, which appears on the dashboard for external applications:

myExternalApp

Owner: isiqAdmin

Product Dashboard

Topic Prefix

isiqAdmin.myExternalApp.EXTAPP.*

Regenerate API Token 

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJpc2lxQWRtaW4ubXlFeHRlcm5hbEFwcC5FWFRBUFAiLCJpYXQiOiJlE2MTM0MjU2NjF9.rucHOS1iFP-I8GQkZ8tDsDjuqpcqHK0XqerbpZr_8ro

3) Next, assuming you intend to consume data, subscribe your external application to an ISIQ source. But suppose you omit this step and immediately call the ISIQ REST API by using the generated API token in a curl command?

```
$ curl -H "Authorization: Bearer  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJpc2lxQWRtaW4ubXlFeHRlcm5hbEFwcC5FWFRBUFAiLCJpYXQiOiJlE2MTM0MjU2NjF9.rucHOS1iFP-I8GQkZ8tDsDjuqpcqHK0XqerbpZr_8ro"  
https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person  
{ "status": 403, "message": "You are not authorized to produce to/consume from this topic." }
```

The API fails with a 403 error. To gain API access to ISIQ topics, you must

- Specify the correct API token string, and
- Subscribe your external application to the ISIQ source that publishes those topics.

Let's try it again after you subscribe MyExternalApp to isim123:

Add a subscription for myExternalApp

☒ Subscribe directly to the data source (no transformation)
☐ Subscribe to the transformed data source (default transformation)
☐ Subscribe to the custom transformed data source (custom transformation)

Select a source to subscribe to:

isiqAdmin.isim123.ISIM 

Subscribe

Cancel

Now a REST call issued by your curl command can retrieve isim123 topic data:

```
$ curl -H "Authorization: Bearer  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJpc2lxQWRtaW4ubXlFeHRlcm5hbEFwcC5FWFRBUFAiLCJpYXQiOiJlE2MTM0MjU2NjF9.rucHOS1iFP-I8GQkZ8tDsDjuqpcqHK0XqerbpZr_8ro"  
https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person
```

```
{["topic":"isqAdmin.isim123.directory.Person","key":"isqAdmin.isim123:erglobalid=00000000000000000007,ou=0,ou=people,erglobalid=00000000000000000000,ou=org,dc=com","value":{"orgname":"Organization","modifytimestamp":"20161119013612.398332Z","audio":null,"businesscategory":null,"carlicense":null,"departmentnumber":null,"displayname":null,"employeeunber":null,"employeeetype":null,"givenname":["Administrator"],"homephone":null,"homepostaladdress":null,"initials":null,"jpegphoto":null,"labeleduri":null,"mail":null,"ersupervisor":null,"manager":null,"mobile":null,"o":null,"pager":null,"photo":null,"preferredlanguage":null,"roomnumber":null,"secretary":null,"uid":["itimadmin"],"usercertificate":null,"userpkcs12":null,"usersmimertificate":null,"x500uniqueidentifier":null,"eracl":null,"erauthorizationowner":null,"erglobalid":"00000000000000000000","erisdeleted":null,"erlastmodifiedtime":"201611190136Z","erlifecycleenable":null,"erparent":{"erglobalid=000000000000000000000000,ou=org,dc=com","erprofilename":null,"eruri":null,"objectclass":["inetOrgPerson","top","person","organizationalPerson","erManagedItem","erPersonItem"],"destinationindicator":null,"facsimiletelephonenumber":null,"internationalisdnumber":null,"l":null,"ou":null,"physicaldeliveryofficename":null,"postaladdress":null,"postalcode":null,"postofficebox":null,"preferreddeliveymethod":null,"registeredaddress":null,"st":null,"street":null,"teletextterminalidentifier":null,"telexnumber":null,"title":null,"x121address":null,"cn":["SystemAdministrator"],"eraliases":null,"ercreatedate":null,"erimageuri":null,"erlastcertifieddate":null,"erlastoperation":null,"erlaststatuschangedate":null,"erlocale":null,"erpersonpassword":null,"erpersonstatus":"0","erpswdlastchanged":"201611190136Z","erroleassignments":null,"errolerecertificationlastaction":null,"errolerecertificationlastactiondate":null,"erroles":["erglobalid=0000000000000000000001,ou=roles,erglobalid=00000000000000000000,ou=org,dc=com"],"ersharedsecret":null,"ersynchpassword":"mekuxT+J9EsWbMcA1GVZsQ==","ercustomdisplay":["Administrator"],"sn":["Administrator"],"description":null,"seealso":null,"telephonenumber":null,"userpassword":null},"partition":1,"offset":0},{["topic":"isqAdmin.isim123.directory.Person","key":"isqAdmin.isim123:erglobalid=6578154610713610604,ou=0,ou=people,erglobalid=0000000000000000000000,ou=org,dc=com","value":{"orgname":"Organization","modifytimestamp":" (and other Person records...)}}
```

In this excerpt, you see how topic data arrives in JSON format. Your application needs to parse the JSON stream to extract the fields of interest, and poll for new data.

Subsequent REST calls might return [], an empty JSON array, if there's no new Person data to consume. That's because Kafka maintains "offsets" in its logs. Offsets are sequential IDs that uniquely identify each record in a Kafka partition. If your application has consumed all 100 Person records, your next REST API calls won't return data until the Person topic has a new Kafka offset that includes Record #101.

If a signed certificate is required to communicate with ISIQ, you need to obtain a copy of the cert file from your security administrator. In that case, the curl command is:

```
$ curl --cacert <your-cert-filename> --H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJpc2lxQWRtaW4ubXlFeHRlcm5hbEFwcC5FWFRBUFAiLCJpYXQoIjE2MTM0MjU2NjF9.rucHOS1iFP-I8GQkZ8tDsDjuqpcqHK0XqerbpZr_8ro" https://ISIQ_hostname/api/external/consume/isqAdmin.isim123.directory.Person
```

If instead of curl you use a REST client program such as Postman, specify the following headers in your GET command:

GET	https://localhost/api/external/consume/isqAdmin.isim123.directory.Person		
Params	Authorization	Headers (2)	Body
▼ Headers (2)			
	KEY	VALUE	DESCRIPTION
☒	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJpc2lxQWRtaW4ubXlFe...	
☒	Content-Type	application/json	

The Authorization header must contain the Bearer token you saved during Step 2 when you configured the external application. The Content-Type header must be set to "application/json".

Query Parameters for Consume API

To give your application more granular control over how many topic records it reads and from which starting offset, the consume API supports the following query parameters:

fromBeginning

The “fromBeginning” parameter can be used to query records from the start of a topic. As mentioned earlier, Kafka maintains offsets in its logs to keep track of the last record read. In some cases, especially during application development, you might want to repeatedly read from a topic’s first record rather than default to the last offset. If so, append the “fromBeginning” query parameter to your consume API call, as shown here:

https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person?fromBeginning

The presence of this parameter causes the consume API to seek to the beginning of the topic. “fromBeginning” must be specified by itself without any associated value. Assigning a value to it throws an error:

```
{
  "status": 500,
  "message": "Query parameter 'fromBeginning' must not have a value"
}
```

Note: “fromBeginning” is the reason the “Reprocess” button (see the “Reprocessing Topics” section) is N/A for external applications that subscribe to a product. Since external applications always have the ability to replay topics from the beginning, there’s no need to offer a “Reprocess” option.

fromRecord=<record_#> (must be a positive integer)

The “fromRecord” parameter can be used to query data from a particular offset in a topic. For example, to read all records starting with the fifth record, construct your API call as follows:

https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person?fromRecord=5

Specifying “fromRecord=1” is functionally equivalent to “fromBeginning”. The “fromBeginning” and “fromRecord=<record_#>” parameters cannot both be used in the same query. Doing so throws an error:

```
{
  "status": 500,
  "message": "Only one of query parameters 'fromBeginning' and 'fromRecord' is allowed"
}
```

Supplying a <record_#> value that’s larger than the total number of records in the topic also throws an error:

```
{
  "status": 500,
  "message": "Value of query parameter 'fromRecord' cannot be larger than the total number of records in the topic"
}
```

maxResults=<max_#_of_records_to_read> (must be a positive integer)

The “maxResults” parameter can be used to limit the results returned by the consume API. Topic logs can grow quite large for products such as IBM Security Identity Manager that potentially load significant amounts of data into topics. To restrict output volume in your consume API calls, assign a

<max_#_of_records_to_read> value. It causes the API to read that number of records starting from the last offset, for example:

https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person?maxResults=30

If the value of “maxResults” is larger than the number of remaining records to be read, all of the remaining records will be returned.

You can also use “maxResults” in conjunction with “fromBeginning” or “fromRecord”, for example:

(1) Return a maximum of 10 records, starting from the beginning of the topic (in other words, from the first record):

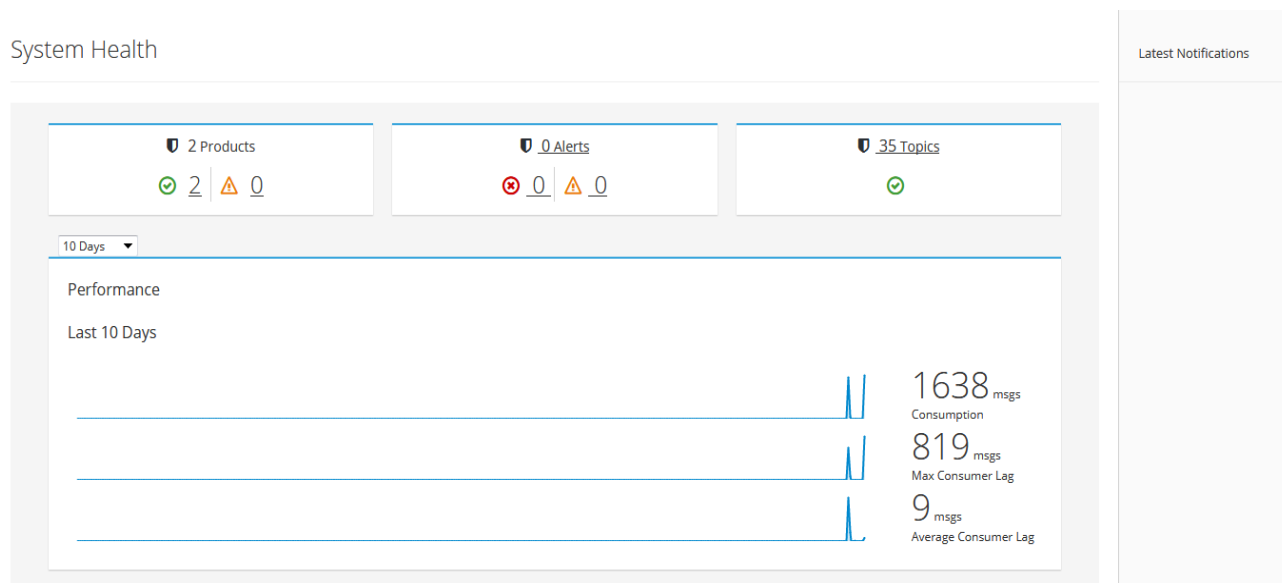
https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person?fromBeginning&maxResults=10

(2) Return a maximum of 20 records, starting from the fifth record:

https://ISIQ_hostname/api/external/consume/isiqAdmin.isim123.directory.Person?fromRecord=5&maxResults=20

Metrics and Alerts

When you log in to ISIQ after at least one product is configured, you are directed to the System Health dashboard, as shown in this sample screen capture:



The dashboard displays three line graphs with statistics about messages consumed by subscribers with your subscriber ID. By default, the graphs cover the last 10 days, grouped into 30-minute slices. You can select the “10 Days” drop-down and choose a different time range, anywhere from 1 Hour to 1 Year. The rightmost data point on each graph is the value recorded in the latest timeslice. When you choose a different time range, the graph automatically adjusts to a corresponding timeslice duration and builds (on a best-effort basis) a collection of data points for the new timeslice. For 1 Hour, the timeslices are 30 seconds. For 1 Day, 5 minutes; 10 Days = 30m; 30 Days = 2h; 6 Months = 12h; 1 Year = 1d.

The dashboard statistics shed light on the data flow between your producers and consumers. Under normal conditions, the Consumption and Lag numbers will surge after a new subscription is configured,

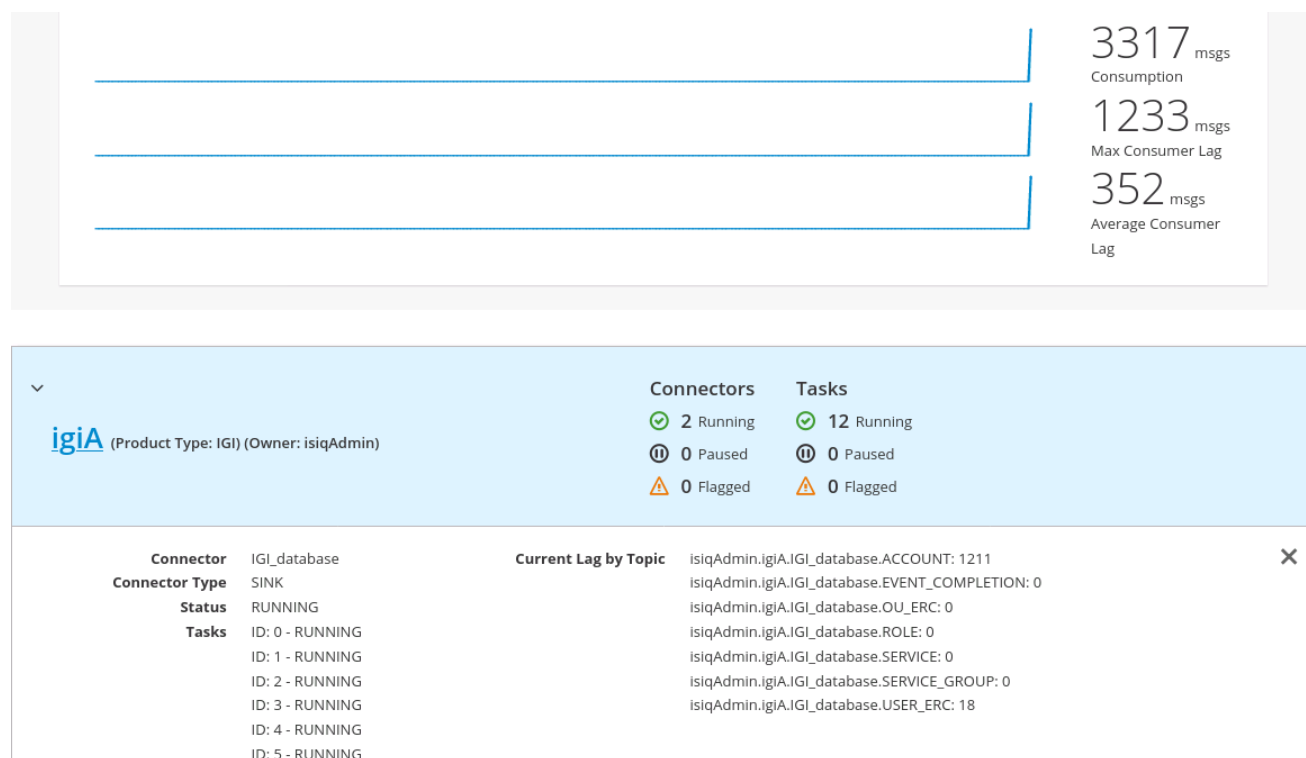
and then the numbers will gradually decrease toward 0. This simply means the consumer is catching up with the producer. Here are the formulas used by ISIQ to calculate the three dashboard statistics:

(1) “Consumption” represents the current Kafka offset minus the previous offset for all topics, summed over the given time range (e.g., last hour).

(2) “Max Consumer Lag” is the sum of the max lags per topic over the given time range.

(3) “Average Consumer Lag” is the sum of the max lags, divided by the number of data points, summed over the given time range.

If the Max Consumer Lag count is fairly high, it means there a lot of messages in the sink topics that have not yet been processed. If you want to drill down on what those unprocessed messages are, expand the subscribing product (“igiA” in this screen capture) to view the “Current Lag by Topic” statistics:



When gauging how much work remains for a subscriber to catch up, the “Current Lag by Topic” statistics provide the clearest picture. They give a topic-by-topic breakdown of the “Max Consumer Lag” value.

There might be cases where dashboard statistics signify a problem:

1. If the “msgs Consumption” number stays at 0 even though you configured a subscription, it likely indicates a product registration problem such as a database connection error in either the producer or consumer. In that case, work with the ISIQ system administrator to view docker service logs to troubleshoot.
2. If the “msgs Consumption” number increased for a period of time, but then remains stuck on the same value, that indicates perhaps the producer or consumer stopped running.
3. If the “Consumer Lag” numbers keep growing and never drop down toward 0, it suggests an issue in the sink connector, which might need to be restarted.

For more information about common problems and their remediation, see the [ISIQ Troubleshooting Guide](#).

The dashboard also shows a count of configured products, topics, and alerts, along with green check marks, yellow warning triangles, and red x's to summarize status. The alert messages inform you of ISIQ system problems, or of actions you need to perform. As an illustration, when you first configure products but do not yet have a subscription defined, a warning alert appears as shown here:

System Health

1 Products

1

0

1 Alerts

0

1

2 Topics

1

Latest Notifications
[Clear Messages](#)

10/13/2022, 9:30:30 AM - WARN [Clear](#)
Unable to retrieve data from metrics database. Verify all services are running, and at least one subscription has been defined.

There is no automatic clearance of alerts. To remove them, you must either click the “Clear” link next to a particular alert or click the “Clear Messages” link to erase all alerts from your dashboard.

Where possible, ISIQ suppresses redundant alerts. For instance, after a failed connection to a product database generates an alert, any failed connection retries to that database do not generate new alerts. There are also configuration options such as ALERT_LAG_MIN (see the “connect-stack.yml” section in Appendix A of the [ISIQ Deployment Guide](#)) that help you manage the frequency of alerts.

The lower half of the System Health dashboard includes a task breakdown for each configured product. In this screen capture, the IGI_Test product was expanded to confirm that all of its tasks are running:

IGI_Test

(Product Type: IGI) (Owner: isiqAdmin)

Connectors

2 Running

0 Paused

0 Flagged

Tasks

12 Running

0 Paused

0 Flagged

Connector

IGISource

Connector Type

SOURCE

Status

RUNNING

Tasks

ID: 0 - RUNNING

ID: 1 - RUNNING

Connector

IGI_database

Connector Type

SINK

Status

RUNNING

Tasks

ID: 0 - RUNNING

ID: 1 - RUNNING

ID: 2 - RUNNING

ID: 3 - RUNNING

ID: 4 - RUNNING

ID: 5 - RUNNING

ID: 6 - RUNNING

ID: 7 - RUNNING

ID: 8 - RUNNING

ID: 9 - RUNNING

Current Lag by Topic

isiqAdmin.IGI_Test.IGI_database.ACCOUNT: 0

isiqAdmin.IGI_Test.IGI_database.DELETES: 0

isiqAdmin.IGI_Test.IGI_database.EVENT_COMPLETION: 0

isiqAdmin.IGI_Test.IGI_database.OU_ERC: 0

isiqAdmin.IGI_Test.IGI_database.ROLE: 0

isiqAdmin.IGI_Test.IGI_database.SERVICE: 0

isiqAdmin.IGI_Test.IGI_database.SERVICE_GROUP: 0

isiqAdmin.IGI_Test.IGI_database.USER_ERC: 0

isim123

(Product Type: ISIM) (Owner: isiqAdmin)

Connectors

3 Running

0 Paused

Tasks

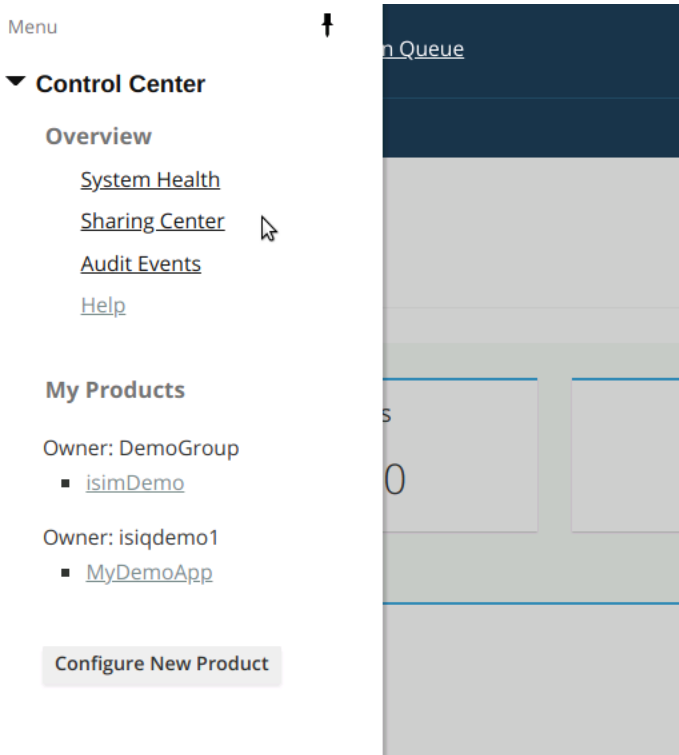
30 Running

0 Paused

31

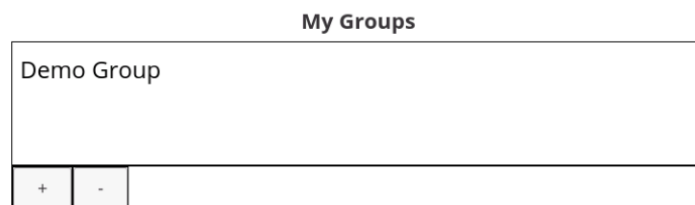
Group Management

If you are not a member of an ISIQ group, it means that you only have access to ISIQ entities whose name begins with your subscriber ID. That limitation might be adequate in certain scenarios, but generally it's beneficial to belong to an ISIQ group comprised of users who share a common set of products and subscriptions. To create and manage ISIQ groups, click the "Sharing Center" link in the navigation sidebar:



In this example, I'm logged in as user "isiqdemo1", and I am a member of a group called "Demo Group", as shown in the following screen capture:

ISIQ Sharing Center



If I highlight "Demo Group" in the "My Groups" selection box, the lower half of the Sharing Center page displays two other selection boxes, one with a list of users belonging to Demo Group, and one with a list of ISIQ products that Demo Group has access to:

My Groups

Demo Group	
+	-

Group Name:

Description:

Group Members

isiqdemo1	
+	-

Group Products

d4e7a98c.isimDemo isiqdemo1.MyDemoApp	
+	-

This screen tells me that my isiqdemo1 ID is the only member of Demo Group. The group has access to two products: (1) “isimDemo”, which is owned by the Demo Group (the “d4e7a98c” hash value indicates the product is group-owned), and (2) “MyDemoApp”, which my individual ID owns since I configured the product when I was unaffiliated with a group. To add or remove group members and products, click the ‘+’ or ‘-’ buttons under the particular selection box, and then click the “Save” button.

If I am an ISIQ user who does not belong to a group, I will see this screen in the Sharing Center:

My Groups

<i>You are not currently a member of any groups. Click the '+' button to create a group.</i> <i>Note: An ISIQ user can belong to only one group.</i>	
+	-

To create a group, I click the ‘+’ button, enter a Group Name and an optional Group Description:

My Groups

You are not currently a member of any groups. Click the '+' button to create a group.

Note: An ISIQ user can belong to only one group.

+

-

Group Name:

Admin

Description:

ISIQ Administrators

Group Members

isigAdmin

+

-

Group Products

isigAdmin.myExternalApp

+

-

--Select a Product-- ▼

Add

Save

Cancel

Since I'm creating a new group, my ID is the only user in the "Group Members" box. I then add one or more products to the Group Products list by clicking the "--Select a Product--" dropdown and the "Add" button. After clicking "Save", I see that the new group was created:

My Groups

Admin

+

-

✓

Admin created successfully

At a later time, I can return to the Sharing Center to add my teammates to the Group Members list, or to modify the list of Group Products.

A key benefit of the Sharing Center is that it decouples a product's subscriber ID from the users who can access the product. For example, if you initially deploy ISIQ with OIDC disabled (see "Shortcuts for Rapid ISIQ Setup" in the [ISIQ Deployment Guide](#)), the generic "isiguser" is the subscriber ID. Every product that isiguser configures, while not a member of a group, has "isiguser" as its first-level qualifier. After you enable OIDC and allow new users to log in to ISIQ, the previously configured product can be shared. As long as isiguser defines a group which includes its configured product and also includes the new OIDC user IDs, those new users can access the product that isiguser configured.

In this scenario, although a product configured by isiquser is later shared with a group, the product's first-level qualifier remains "isiquser". A product's name never changes throughout its lifecycle, no matter if it is later shared or unshared between groups. If you prefer that full product names should always start with a group name, then isiquser must define a group with itself as a member *before* configuring any products. That will put the group name as the first-level qualifier.

Usage Notes:

- If you highlight the group in the "My Groups" selection box and click the '-' button, you are asked to confirm whether you want to delete the group. If you click the "Delete" button, the group is deleted. But note that any products which were assigned to the group are NOT deleted. Those products still exist in ISIQ and can be assigned to another group.
- In pre-ISIQ 1.0.7 releases, groups were known as "teamlists", and their members were defined in <starterKitDir>/cfg/rest/teamlist.json. As of ISIQ 1.0.7, teamlist.json has been replaced by an in-memory groups registry, which is backed by ISIQ volumes. If you have a teamlist.json file from an earlier ISIQ release, the file's contents will be read and stored in the groups registry after you upgrade to ISIQ 1.0.7 (or later) and start it for the first time. From that point on, the teamlist.json file is no longer used.
- Since an ISIQ user can belong to only one group, it requires a two-step process to move a user between groups. Suppose you are a member of GroupA, and a member of GroupB wants to add you to GroupB. The add attempt will fail. You must first be removed from GroupA before a member of GroupB can successfully add you.
- If you make a change in the Sharing Center UI, reverse the change, and click the "Save" button, the update fails. For example, if you remove ProductA from Group1 using the '-' button, and then you decide to re-add ProductA and click "Save", there is no net change to Group1. In that case, you see the message: "Failed to update Group1. Group with the specified configuration already exists". This is an expected message you can ignore. To avoid cluttering the groups registry with redundant entries, ISIQ checks that at least one change was made to the previously saved group entry. If there are no changes, ISIQ does not write out the same entry.

Custom Transformations

ISIQ provides a default template file, txdef.json, that defines how attributes are transformed when one product subscribes to another. These default transformation rules are sufficient for most product integration scenarios. However, if you need attributes to be consumed in a specialized format, you can modify txdef.json to suit your customization requirements.

Recommendation: Always back up txdef.json before modifying it. This is important because you cannot use an alternate name or location for the file. It must be named txdef.json, and it must reside in the cfg/connect directory as referenced in ISIQ's app-stack.yml and connect-stack.yml files.

Transformation rules are written in JSON format. ISIQ uses the JSON Query (JQ) package as its JSON processor. To illustrate the transformation syntax, here are excerpts from the default txdef.json:

```
[
{
  "in": "isim",
  "out": "igi",
  "smts": [
    {
```

```

    "in": "Person",
    "out": "USER_ERC",
    "txe": "{pm_code: .uid[0], ou: .erparent, given_name: .givenname[0], surname: .sn[0], email: .mail[0], address:
.postaladdress[0], erroles: .erroles}"
  },
  {
    "in": "BPPerson",
    "out": "USER_ERC",
    "txe": "{pm_code: .ercustomdisplay[0], ou: .erparent, given_name: (.cn[0]|split(\" \")|. [0]), surname: .ercustomdisplay[0],
email: .mail[0], erroles: .erroles}"
  },
  ...
},
...
{
  "in": "igi",
  "out": "isim",
  "smts": [
    {
      ...
    },
  ],
}
]

```

The “in” and “out” keys specify the product type: source/producer products are “in” and sink/consumer products are “out”. Single Message Transformations (“smts”) describe mappings between source and sink entity topics. Within the first “smts” block, “Person” and “BPPerson” are ISIM entity topics that are transformed and output to the USER_ERC topic for IGI to consume.

The “txe” key defines the JQ filters that get applied before attribute mappings and assignments take place. A “txe” can be specified as a null object “{}” if no transformations are necessary. If an attribute can have multiple values, it uses an array format, for example, .uid[0]. For more information about JQ syntax and the JQ package, see <https://stedolan.github.io/jq/manual/>.

To supplement the JQ package, ISIQ also includes the jackson-jq and jackson-jq-extra packages. These two open source libraries provide additional JQ filters that might aid you in your transformation logic. To learn more, see <https://devhub.io/repos/eiiches-jackson-jq>. A summarized list of their offerings can be found in the “Language Features / Functions” table at <https://github.com/eiiches/jackson-jq/blob/develop/1.x/README.md>.

The ISIQ-supplied transformation template supports two built-in acronyms:

1. “isim”: (IBM Security Identity Manager)
2. “igi”: (IBM Security Verify Governance)

These acronyms may be used as “in” or “out” fields when defining custom transformations for your source and sink products.

A custom transformation can be created before a new subscription is processed, or it can be applied to an existing subscription by reprocessing the relevant topic data (see “Reprocessing Topics” for details).

Usage Notes:

- If you add entity topics (for example, “Person” and “USER_ERC” in the excerpt above) to txdef.json, it requires a restart of ISIQ’s app and connect stacks for the changes to take effect.

Also, you might have to update your ISIQ subscriptions (refer to "Appendix D: Updating Subscriptions") so that consumers are aware of the new topics.

- In case of duplicates in the product-to-product or entity-to-entity mappings, ISIQ honors the first occurrence of a custom transformation, i.e., the one earlier in the txdef.json file.
- C/C++ style comments (`/* ... */` and `//`) are supported inside the file.
- Before implementing txdef.json changes, run the file through a JSON validator tool (there are many available on the web). A stray comma or other minor syntax error will cause ISIQ to use the default txdef.json instead of your customized version. This can have serious downstream consequences as far as expected data transformations not occurring.
- Before upgrading to a newer ISIQ release, be sure to save a copy of your customized txdef.json. You will want to retrofit your file changes after the upgrade.
- For considerations when customizing ISIM-to-IGI transformations, see "Appendix E: Custom ISIM-to-IGI Transformations".

OpenShift Container Platform Instructions:

If you're using the OpenShift Container Platform (OCP), the contents and location of txdef.json are the same as with Docker Swarm. However, the steps for implementing txdef.json changes are as follows:

(a) Run createConfig.sh:

```
<starterKit>/openshift/createConfig.sh txdef.json
```

This creates a new 051-config-txdef.yaml in the <starterKit>/openshift/templates directory.

(b) Run an `oc apply` command against the YAML representation of txdef.json:

```
oc apply -f yaml/txdef.yaml
```

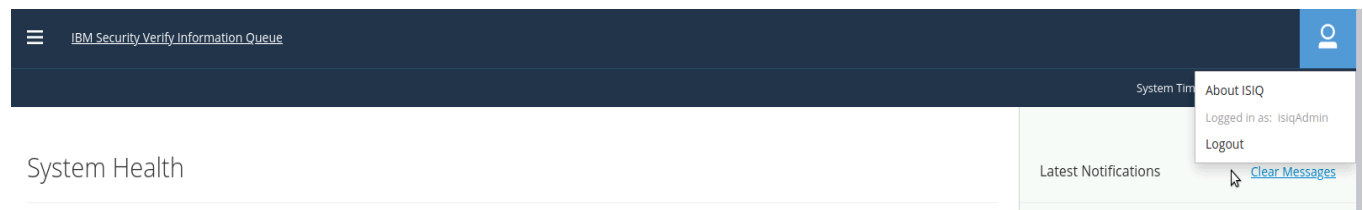
This loads the new txdef.json data into OCP.

(c) Delete the isiq-connect pods so that they will be recreated with the modified txdef.json.

For more information, refer to "Implementing configuration file changes" in the [ISIQ Deployment Guide](#).

Logging Out

After you configure products and verify that your producers and consumers are working, you can log out of ISIQ by clicking the user avatar in the upper right and selecting the "Logout" menu option:



At any later time, you can log back in to ISIQ to edit information about your existing products, configure new products, set up more subscriptions, or check the System Health dashboard.

If your ISIQ session is idle for 60 minutes, you're automatically logged out. You can adjust the automatic timeout by changing the environment variable, `ISIQ_SESSION_INACTIVITY_TIMEOUT_IN_MINUTES=60`, in the app-stack.yml file.

Appendix A: IGI Customizations for ISIQ

Here are the IGI customizations which must be made to support ISIQ integration:

Define adequate temporary table space size

The temporary table space size in IGI is typically set to 'M' for Medium. During ISIQ deployments involving large-scale loads of ISIM records into IGI, the 'M' setting can result in errors such as:

DM6017E The following table space is full. Table space name: "TEMPSPACE1".

Table space identifier: "1". Container path:

"home/igiinst/igiinst/NODE0000/IGIDB/T0000001/C0000000.TMP". Container identifier "0"

To avoid these types of out-of-space errors, we recommend defining the TABLESPACE_SIZE as 'L' for Large during installation of the IGI DB, or when upgrading the DB.

For other IGI recommendations regarding database configuration matters, refer to the IBM

Documentation link: [Managing the database server configuration](#).

Review ISIM customizations

NOTE: Because ISIM is highly customizable, there is a chance that existing customizations in your ISIM system will cause problems when integrating with IGI. For example, if you greatly expanded the maximum ISIM group name length, ISIQ cannot insert those extra-long group names into IGI if the default IGI column size isn't large enough. To address these sorts of scenarios, you can:

1. Perform a "dry run" of ISIM-to-IGI integration in a test environment. Check for any alerts on the ISIQ System Health dashboard, or errors and exceptions in the ISIQ docker logs, which indicate a data type or size incompatibility between ISIM and IGI.
2. After the dry run completes, run the ISIM-to-IGI validation tool (for more information about the tool, see "Appendix H: Optional ISIQ Tools"). The validation tool outputs a series of messages as it compares your ISIM LDAP directory against your IGI database. Any unintegrated ISIM entities are flagged, which can help you identify customized ISIM data that needs follow-up action.

If you find that a column in an IGI table isn't large enough to hold a customized ISIM attribute, you can resolve it by using IGI's DATA_TRANSFORMATION_MAPPING table. For more information about this table, see "Diagnosing Database Problems" in the [ISIQ Troubleshooting Guide](#).

Enable the EVENT_OUT_USER table

To support user integration from IGI to ISIM, ISIQ relies on events in IGI's EVENT_OUT_USER table. There are two steps for enablement:

1. In ISIQ's connect-stack.yml file, set the ISIQ_IGItoISIM_FULFILL_USER_EVENTS environment variable to true. For more information about the connect-stack.yml file, see "Configuring the YAML Files" in the [ISIQ Deployment Guide](#).
2. IGI's user event-out mechanism operates via a database trigger, which directs the USER_ERC table to add an event to the EVENT_OUT_USER table each time an IGI user is created, modified, or deleted. ISIQ takes these events and sends them to your configured ISIM product.

The database trigger, named TRG_USER_ERC_TO_OUT, is disabled by default. To enable it, extract the contents of <starterKitDir>/util/IGIUserChangeLogTrigger.zip. Inside the .zip file, you will find a README and a set of .sql files for defining the EVENT_OUT_USER table (it should already exist in your IGI

database), and for enabling and disabling TRG_USER_ERC_TO_OUT. Depending on your IGI database type, run one of the following commands to enable or disable the trigger:

DB2)

Enable:

```
clppplus -nw igacore/<passwd>@<host>:<port>/<db-name> @en-db2.sql
```

Disable:

```
clppplus -nw igacore/<passwd>@<host>:<port>/<db-name> @dis-db2.sql
```

Oracle)

Enable:

```
sqlplus iga_core/<passwd>@<SID> @en-ora.sql
```

Disable:

```
sqlplus iga_core/<passwd>@<SID> @dis-ora.sql
```

PostgreSQL)

Run psql commands (note: you will be prompted for the iga_core password):

Enable:

```
$PG_HOME/bin/psql --host=<host> --port=<port> --dbname=<db-name> --username=iga_core --  
file=en-pg.sql
```

Disable:

```
$PG_HOME/bin/psql --host=<host> --port=<port> --dbname=<db-name> --username=iga_core --  
file=dis-pg.sql
```

OR

Use a PostgreSQL GUI package such as pgAdmin:

Log in as iga_core

Enable:

a) Open query panel and open file en-pg-pgadmin.sql

b) Execute query

Disable:

a) Open query panel and open file dis-pg.sql

b) Execute query

For customers who used the ISIGADI product, a similar trigger enablement procedure is documented at:

https://www.ibm.com/support/knowledgecenter/en/SSGHJR_5.2.6/com.ibm.igi.doc/administering/cpt/cpt_isigadi_triggers.html.

Update the USER_ERC attribute mapping to set the user DN

The default IGI does not allow the USER_ERC table to set a person's DN field. Therefore, you need to update the USER_ERC table attribute mapping so that a person's DN field can be set with the key (productPrefix.personDn).

The screenshot shows the IBM Identity Governance and Intelligence interface. The top navigation bar includes 'Identity Governance and Intelligence', 'Access Governance Core', 'Ideas / admin', 'Help', 'Logout', and the IBM logo. Below this is a secondary navigation bar with 'Manage', 'Configure', 'Monitor', 'Tools', and 'Settings'. The 'Settings' tab is active, showing 'Core Configurations' and 'Configure Password Service'. Under 'Core Configurations', the 'User Virtual Attributes' sub-tab is selected. The interface is split into two panels. The left panel, titled 'Repository', contains a table with columns 'Ena', 'Name', and 'Type'. It lists 'UserErc' and 'S_User'. The right panel, titled 'Details' with a sub-tab 'Attribute Mapping', contains a table with columns 'Name', 'Label', and 'Lookup'. It shows 'ATTR11' with label '_DN' and 'ATTR10' with label 'ATTR10'. Buttons for 'Filter', 'Actions', 'Save', 'Cancel', and 'Actions' are present.

Ena	Name	Type
☒	UserErc	De
☒	S_User	De

Name	Label	Lookup
☒ ATTR11	_DN	
☐ ATTR10	ATTR10	

1. Go to Access Governance Core -> Settings -> Core Configurations -> User Virtual Attributes page.
2. Select UserErc from the list on the left.
3. Go to the right panel and select Attribute Mapping and Actions -> Add.
4. Select ATTR11 and click OK.
5. Change the label to _DN, which will set the value of ATTR11 from USER_ERC to the DN column of the PERSON table. The underscore character is used for the PERSON table column.
6. Select Save to record the change.

Update the Create Account rule flow

Issues: The Create Account rule flow does not allow the user DN to match the owner when creating an account. It also does not allow the account DN to be set to the account.

The Live Events -> TARGET -> Create Account rule flow needs to be updated with the new matching rule (note: "Create Account" was known as "ACCOUNT_CREATE" prior to IGI 5.2.6). EVENT_TARGET.ATTR8 is leveraged to get the user DN and it will be used to match the account owner. EVENT_TARGET.DN is leveraged for setting the account DN and it will be stored in the PWDMANAGEMENT.DN.

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> TARGET -> Create Account.
3. Expand the Rules Package section.
4. Select "Create Account—userDn Matching Rule".
5. Add the "Create Account – userDn Matching Rule" rule to the "Run" list as shown here (note: this screen capture is from an IGI 5.2.5 FP1 system that was using the "ACCOUNT_CREATE" rule flow name):

Rules Rule Flows

Rule Class: Live Events

Queue: TARGET

Rule Flow: ACCOUNT_CREATE

Before Run After

ACCOUNT_CREATE

- Create Account Target Default Group
- Check level
- Create Account - userDn Matching Rule
- Create Account [Email Matching]
- Create Account [Userid Matching]
- Create Account [Post Matching]

Rule Concept

Rules Package

Name	Description
[EXAMPE] Create Account - custom matching	[V1.1 - 2014-05-26] - Match the user using the description fr
Check level	[V1.0 - 2014-05-26]
Create Account - userDn Matching Rule	[V1.1 - 2019-05-03]
Create Account [Email Matching]	[V1.5 - 2014-05-26] - email = event.getAttr3()
Create Account [Name-Surname Matching]	[V1.5 - 2014-05-26] - name = event.getAttr1() surname = event.getAttr2()
Create Account [Post Matching]	[V1.5 - 2014-05-26]
Create Account [Userid Matching]	[V1.5 - 2014-05-26]
Find account attributes	[V1.0 - 2016-09-30]
Update Target Account with Sync Password	[V0.1 - 2018-03-23]

Items per page: 50 | Results: 9 | 1 of 1

Package Imports

- Select "Create Account Target Default Group" in the Run list.
- Select the "Create Account – userDn Matching Rule" rule in the Rules Package.
- Go to Actions -> Add.

6. Make sure the rule is positioned in the Run list before the "Create Account [Email Matching]" rule (use the "Move Up" Action as needed), which is illustrated in the screen capture.

Update the Create Org. Unit rule flow

Issue: The default Create Org. Unit rule flow doesn't use the ORGANIZATIONAL_UNIT_ERC table to set the owner when creating an OU.

The Create Org. Unit rule flow needs to be modified to use the ORGANIZATIONAL_UNIT_ERC.ATTR1 value to set the owner (note: "Create Org. Unit" was known as "ORGUNIT_ADD" prior to IGI 5.2.6).

- Go to Access Governance Core -> Configure -> Rules.
- Select Live Events -> IN -> Create Org. Unit.
- Expand the Rules Package section.
- Select "Add OU with owner".
- Replace the existing "Add OU" rule with the "Add OU with owner" rule under "Create Org. Unit -> Add OrgUnit default group".

For more information about rule replacement, see "How to replace an existing rule with a new rule" at the end of this Appendix.

Update the Create User rule flow

Issue 1: Since the user creation event and the account creation event are handled from different IGI event queues, an account can be created before the corresponding user is created. When this happens, the account is created as an orphan. After the user gets created, the account must be reassigned to the owner from the IGI UI. To resolve this issue automatically, a new rule needs to be created and assigned to the "Create User" rule flow (note: "Create User" was known as "USER_ADD" prior to IGI 5.2.6).

Issue 2: The default password, which is the same as the user ID, cannot be specified when creating a user. The generated password must be used.

- Go to Access Governance Core -> Configure -> Rules -> Live Events -> IN -> Create User.

2. Expand the Rules Package section, select "Assign Orphan Accounts Using User Dn".
3. Add the rule to the Create User flow.
 - a. Select "Add User Default group".
 - b. Select the "Assign Orphan Accounts Using User Dn" rule.
 - c. Actions -> Add.
4. Replace "Set UserID Password" with "Set Random Password" from "Add User default group" (note: this step became unnecessary beginning with IGI 5.2.5 FP1. However, you might still need to update the "Set Random Password" rule. Refer to https://www.ibm.com/support/knowledgecenter/SSGHJR_5.2.6/com.ibm.igi.doc/administering/tsk/tsk_ac_change_set_random_password_rule.html).
5. Enable password notification so that the user will be notified of the generated password. Only implement this step if you want to send out password notifications for the IGI system account.
 - a. Go to Manage -> Account Configurations.
 - b. Select Ideas. The Ideas account is the IGI system account.
 - c. From the Creation Policy tab, check Notification on account creation.
 - d. Save.

Update the Modify OrgUnit rule flow

Issue: By default, IGI does not allow ISIQ to set, replace, or remove the owner of an Organizational Unit. To support these capabilities, you must add a new rule, "Modify OrgUnit with Owner", as described in the following steps. Note: these steps assume IGI 5.2.6 or higher:

1. Go to Access Governance Core -> Configure -> Rules -> Live Events -> IN -> Modify Org. Unit.
2. Expand the Rules Package section.
3. In the IGI text editor, create a rule named "Modify OrgUnit with Owner" with this content:

```
when
  event : EventInBean( )
  orgUnitBean : OrgUnitBean( )
  orgUnitErcBean : OrgUnitErcBean( )
then
  // obtain and assign the ownerCode
  String ownerCode = (String)orgUnitErcBean.getAttribute("ATTR1");
  if (ownerCode != null && ownerCode.trim().length() != 0) {
    orgUnitBean.setOwnerCode(ownerCode);
    orgUnitBean.setPerson_code(ownerCode);
  } else {
    orgUnitBean.setOwnerCode(null);
    orgUnitBean.setPerson_id(null);
    orgUnitBean.setPerson_code(null);
    orgUnitBean.setPerson_name(null);
    orgUnitBean.setPerson_surname(null);
    orgUnitBean.setPerson_email(null);
  }
  OrgUnitAction.modifyOrgUnit(sql, orgUnitBean);
  logger.debug("OU modified: " + orgUnitBean);
4. Save the "Modify OrgUnit with Owner" rule.
```

5. Replace the existing “Modify OrgUnit” rule with the “Modify OrgUnit with Owner” rule under “Modify Org. Unit -> Modify OrgUnit default group”. Here’s what you should see for the updated rule flow:

The screenshot displays the IBM Security Identity Governance and Intelligence (IGI) Access Governance Core interface. The top navigation bar includes a hamburger menu, the text "IBM Security Identity Governance and Intelligence", and "Access Governance Core". Below this is a secondary navigation bar with tabs: "Manage", "Configure", "Monitor", "Tools", and "Settings". The "Configure" tab is active, and within it, the "Rules" sub-tab is selected. The main content area is titled "Rule Flows" and contains a configuration panel on the left and a rule list on the right.

Configuration Panel (Left):

- Rule Class:** Live Events (dropdown)
- Queue:** IN (dropdown)
- Rule Flow:** Modify Org. Unit (dropdown)
- Actions:** A blue button with a downward arrow.
- Flow Diagram:** A diagram showing the rule flow. It starts with "Before", followed by "Run" (highlighted), and then "After". Under "Run", there is a dropdown menu showing "Modify Org. Unit" expanded, revealing a sub-flow "Modify OrgUnit default group", which then leads to "Modify OrgUnit with Owner".

Rule List (Right):

☐	Name	Desi
☐	Modify OrgUnit	[V1
☐	Modify OrgUnit with Owner	

Set Group DN to entitlement

Issue: By default, IGI does not allow the group DN to be set to the entitlement. To store the group DN in the entitlement table, you need to update the Create Application Access rule flow. You also need to update the Assign Application Access rule flow since a new entitlement will be created if the entitlement isn’t found when assigning a permission or external role to the user.

- Update the Create Application Access rule flow

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> Target -> Create Application Access (note: “Create Application Access” was known as “ROLE_CREATE” prior to IGI 5.2.6).
3. Expand the Rules Package section.
4. Replace the existing “Create a new role” rule with “Create a new role - with Dn” under Create Application Access.

- Update the Assign Application Access rule flow

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> Target -> Assign Application Access (note: “Assign Application Access” was known as “PERMISSION_ADD” prior to IGI 5.2.6).
3. Expand the Rules Package section.

4. Replace “Create permission if it doesn’t exist” with “Create permission with DN if it doesn’t exist” under “Assign Application Access”.

Optional IGI customizations

The following IGI customizations are optional and should only be implemented if you need to support the particular integration feature.

Integrate ISIM orphan accounts into IGI as orphan accounts

By default, ISIQ does not integrate ISIM orphan accounts into IGI as orphan accounts. To enable this functionality, you must make two additional IGI rule flow customizations (note: these instructions assume IGI 5.2.6 or above).

- Update the Create Account rule flow

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> TARGET -> Create Account.
3. Under “Create Account Target Default Group”, remove the “Create Account [Email matching]” and “Create Account [Userid Matching]” rules.
4. Expand the Rules Package section.
5. Select “Create Account—userDn Matching Rule” and choose Actions -> Modify.
6. From the text editor, find the following block of code at the beginning of the rule:

```
// Exit if account owner dn is null
if (event.getAttr8() == null) {
    logger.info("Account owner DN is empty. Account is not created!");
    return;
} else {
    // Set owner DN to attr5
    account.setAttr5(event.getAttr8());
}
// If the account dn is set, then set it to the account.
if (event.getDn() != null) {
    account.setDn(event.getDn());
}
```

7. Replace it with this block of code:

```
// If the account dn is set, then set it to the account.
if (event.getDn() != null) {
    account.setDn(event.getDn());
}
// Exit if account owner dn is null
if (event.getAttr8() == null) {
    UserAction.addAccount(sql, account);
    logger.info("Account : " + account.getCode() + " created!");
    return;
} else {
    // Set owner DN to attr5
    account.setAttr5(event.getAttr8());
}
```

8. Save the changes.

- Update the Modify Account rule flow

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> TARGET -> Modify Account.
3. Expand the Package Imports section and insert the following import statement if it's not present:
import com.engiweb.profilemanager.common.ruleengine.action.reorganize._AccountAction
4. Click Save.
5. Expand the Rules Package section.
6. Select Actions -> Create, to add a new rule as shown here:

Name: Handle Orphan Accounts

Description: Update accounts that need to be orphaned or adopted

when

 event : EventTargetBean()

 account : AccountBean()

then

// [V5.2.6 - 2021-03-23]

BeanList accountList = _AccountAction.findAccount(sql, account, null);

if (!accountList.isEmpty()) {

 account = (AccountBean) accountList.get(0);

 // If the account dn is set, then set it to the account.

 if (event.getDn() != null) {

 account.setDn(event.getDn());

 }

 boolean hasOwner = account.getPerson_id() != null;

 // logger.debug("Orphan Rule: personID = " + account.getPerson_id());

 // logger.debug("Orphan Rule: hasOwner = " + hasOwner);

 if (event.getAttr8() == null) {

 // Orphan account

 if (hasOwner) {

 _AccountAction.unMatchAccount(sql, account);

 }

 } else {

 // Adopt account

 if (!hasOwner || account.getAttr5() != event.getAttr8()) {

 UserBean userFilter = new UserBean();

 userFilter.setDn(event.getAttr8());

 BeanList beanList = UserAction.find(sql, userFilter);

 if (!beanList.isEmpty()) {

 userFilter = (UserBean) beanList.get(0);

 account.setPerson_id(userFilter.getId());

 account.setIsDefault(1L);

 _AccountAction.matchAccount(sql, userFilter, account);

 }

 }

 }

}

7. Save the "Handle Orphan Accounts" rule.
8. From the left panel, select the parent of the "Modify Account" rule.
9. From the Rules Package section, select "Handle Orphan Accounts".
10. Select Actions -> Add.
11. From the Run section, place the new rule under the existing "Modify Account" rule.

Allow the account user ID to be changed for a service account

Service accounts are the non-system accounts that you can create after service profiles are imported. By default, the Modify Account rule flow does not let you change the account user ID for a service account. To enable this functionality, you must customize the Modify Account rule flow (note: these instructions assume IGI 5.2.6 or above).

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> TARGET -> Modify Account.
3. Expand the Package Imports section and insert the following import statement if it's not present:
import com.engiweb.profilemanager.common.ruleengine.action.reorganize._AccountAction
4. Click Save.
5. Expand the Rules Package section.
6. Select Actions -> Create, to add a new rule as shown here:

Name: Modify Account with New User ID

Description: Searches the account with the account DN instead of the account ID (code)

when

 event : EventTargetBean()

 accountBean : AccountBean()

then

logger.debug("===== Modify Account rule is being executed...");

BeanList<AccountBean> acctBeanList = _AccountAction.findAccount(sql, accountBean, null);

if (!acctBeanList.isEmpty()) {

 _AccountAction.updateAccount(sql, accountBean);

} else {

 AccountBean acctFilter = new AccountBean();

 acctFilter.setDn(event.getDn());

 acctBeanList = _AccountAction.findAccount(sql, acctFilter, null);

 if (!acctBeanList.isEmpty()) {

 acctFilter = acctBeanList.get(0);

 acctFilter.setCode(event.getCode());

 acctFilter.setEmail(accountBean.getEmail());

 acctFilter.setName(accountBean.getName());

 acctFilter.setDisplayName(accountBean.getDisplayName());

 acctFilter.setSurname(accountBean.getSurname());

 _AccountAction.updateAccount(sql, acctFilter);

 }

}

logger.info("===== Account modified!");

7. Save the "Modify Account with New User ID" rule.
8. From the left-side panel, select the parent of the Modify Account rule.
9. From the Rules Package section, select "Modify Account with New User ID".
10. Select Actions -> Add.

11. From the Run section, replace the “Modify Account” rule with “Modify Account with New User ID”.
12. Note: If you also added the optional "Handle Orphan Accounts" rule, make sure that "Modify Account with New User ID" is positioned in the Run list before "Handle Orphan Accounts".

Support multiple user sources for IGI

If you have multiple IBM Security Identity Manager sources sending identities to IGI, or you have one or more ISIMs and an HR feed sending identities, you might want to leverage IGI’s User Multiple Entries (UME) feature. With this feature, the first occurrence of an identity loaded into IGI—whether from ISIM or an HR feed—becomes the primary user. Any extra occurrences of that identity are created as UMEs, which are secondary users linked to the primary.

For more information about UMEs, see the IBM Documentation link: [Users, attributes, and polyarchies](#).

How to create a new rule

To create a new rule in IGI, you need to

1. Go to Access Governance Core -> Configure -> Rules.
2. Select the rule flow in which you want to create a new rule. For example, if you want to create a new rule for the Create User rule flow, select Live Events -> IN -> Create User.
3. Expand the Rules Package and select Actions -> Create.

Note: Do not select “Add”. The Add action is for adding an existing rule from IGI’s rules catalog to the rule flow. The catalog includes all available rules that can be used in the selected rule flow. The “Create” action is for coding a new rule in the IGI text editor.

4. Supply the name, an optional description, and the code for the new rule.
5. Click Save.

How to replace an existing rule with a new rule

Let’s assume that you created a rule called “Create a new role with DN” and you now want to replace that rule in the Create Application Access rule flow.

To replace the existing rule with a new rule, you need to

1. Go to Access Governance Core -> Configure -> Rules.
2. Select Live Events -> TARGET -> Create Application Access rule flow.
3. Select the parent entry in which you want to add the rule. For example, select Create Application Access from the Run section of the rule flow.
4. Select the new rule “Create a new role with DN” and select Actions -> Add.
5. Once the new rule is added, it is placed at the bottom. Move the new rule next to the rule that you want to replace. Select the new rule and select Actions -> Move Up.
6. Once the new rule is positioned next to the existing “Create a new role”, select “Create a new role” and select Actions -> Remove.

Appendix B: ISIM Customizations for ISIQ

Here are the ISIM customizations to make in support of ISIQ integration:

Processing ISIM deletes

In order for ISIQ to process deleted ISIM entities such as users, accounts, and roles, the “change log” option in the IBM Security Directory Server for ISIM must be configured. The details for configuring it can be found at:

https://www.ibm.com/support/knowledgecenter/en/SSVJJU_6.4.0/com.ibm.IBMDS.doc_6.4/c_ig_changelog_management.html

To confirm that “change log” is working, you can check the “Topic List” UI, as described earlier in “Connectors and Topics”. Search for the <subscriberID>.<isim-name>.directory.DELETES topic and (assuming you have an IGI subscribed to ISIM) the <subscriberID>.<igi-name>.IGI_database.DELETES topic. In the “Msgs” column for both, you should see a value greater than 0. If that’s not the case, and you have deleted ISIM entities, it probably means that you must update your IGI-to-ISIM subscription (refer to [Appendix D: Updating Subscriptions](#)). If you didn’t delete ISIM entities until *after* you defined the IGI-to-ISIM subscription, then the subscription won’t know about the DELETES topics. Updating the subscription will allow those topics to be detected and processed.

Resetting the Change Log Offset

The change log has a counter that’s incremented whenever add, modify, or delete operations are performed in the directory instance. At the time you configure an IBM Security Identity Manager product in ISIQ, the ISIM directory connector stores the current change log counter, and uses that counter as an offset to know where to begin searching for deleted ISIM entities. If your LDAP is upgraded, or a failover occurs in which the primary LDAP switches to a backup, the change log counter will reset to 1. In that circumstance, ISIQ must also update its own change log offset.

To address this scenario, you should run the <starterKitDir>/util/resetChangelog.sh script. Its purpose is to reset ISIQ’s change log offset so that ISIM deletes will continue to be detected and processed correctly. For example:

```
./resetChangelog.sh admin.myISIM.directory 1000
```

Note: By running the script, you are not modifying the LDAP change log itself. You are simply modifying an internal offset value used by ISIQ to know where to begin searching the change log.

If you have multiple IBM Security Identity Manager products that are affected by an LDAP upgrade or failover, run the resetChangelog.sh script multiple times, once for each configured ISIM. The script contains instructions for how to specify the IBM Security Identity Manager product whose offset you want to reset. After running the script, restart ISIQ to ensure that the reset offset value will be used.

Appendix C: Mapping Entities and Events between ISIM and IGI

In ISIQ, when an IGI product subscribes to an IBM Security Identity Manager product, the various ISIM entities are loaded into corresponding entities in IGI. Conversely, when an IBM Security Identity Manager product subscribes to an IGI product, a series of IGI events are sent to initiate updates in ISIM.

This Appendix lists (1) the mapping relationships between ISIM and IGI entities, (2) the operations (add, remove, delete) that can be performed by one product on the other, and (3) the types of events that flow between the two products.

Entity-to-Entity Mapping

The following table shows how ISIM entities map to IGI entities:

ISIM Entity	IGI Entity
Organization	Organizational Unit
Admin Domain	Organizational Unit
Business Partner Unit	Organizational Unit
Organizational Unit	Organizational Unit
Organizational Role	External Role
Service	Application, Account Configuration, Events Marker
Group	Permission
ISIM System Role (Group in ITIM Service)	Permission
Person	User
Account	Account

For every ISIM organizational role loaded into IGI, a new application and account configuration will be created with the name, “ISIM – <isimProduct>”. For example, if you configured IBM Security Identity Manager product “myISIM6” in ISIQ, and subscribed an IGI product to “myISIM6”, the application and account configuration created in IGI will be named “ISIM – myISIM6”.

For every ISIM person loaded into IGI, if the person has an organizational role, the IGI user will be created with a corresponding external role assigned. Using the example, all organizational roles from myISIM6 will be ported to IGI as external roles in the “ISIM – myISIM6” application.

If an ISIM account has a group, an IGI account will be created under the corresponding account configuration, and a permission for that ISIM group will be assigned to the account owner in IGI.

Entity-Key Field Mapping

For every ISIM entity loaded into IGI, the DN value will be used as the key. ISIQ prefixes the keys with “<subscriberId>.<isimProduct>:”. The following table shows which ISIM key gets used for which ISIM entity, and which IGI field will contain the key:

ISIM Entity	ISIM Key	IGI field where the key is stored
Organization	DN	ID Code
Admin Domain	DN	ID Code
Business Partner Unit	DN	ID Code
Organizational Unit	DN	ID Code
Organizational Role	DN	External Ref
Service	DN	Application: APPLICATION.VALUE Account Configuration: PWDCFG.VALUE Events Marker: TARGET.VALUE In the IGI UI, the service DN is shown as the events marker value
Group	DN	External Ref
ISIM System Group (Group in ITIM Service)	DN	External Ref
Person	DN	DN
Account	DN	DN

For every service in ISIM, an application, account configuration, and events marker will be created in IGI. The same service key is stored in all three entities. However, in the IGI UI you see only the key from the events marker field (note: APPLICATION, PWDCFG, and TARGET are the names of the IGI database tables for application, account configuration, and events marker).

Events Fulfilled for IGI-to-ISIM Integration

ISIQ fulfills the following events for IGI-to-ISIM integration:

Account events:

- Add account
- Lock (suspend) account
- Unlock (restore) account
- Delete account

Entitlement change events:

- Add permission to a user
- Remove permission from a user
- Add ISIM external role to a user
- Remove ISIM external role from a user

User events:

- Create a user

For the account and entitlement change events, the account owner and corresponding service must exist in ISIM. Events for users that don't exist in ISIM will be ignored. Events for targets that don't exist in ISIM are also ignored.

For ISIM roles, an ISIM-<productName> target is created in IGI and the external roles are created in the target. Events for the target are fulfilled by assigning the corresponding role to the user in ISIM.

The “Create User” event will be processed by ISIQ if the following conditions are met:

1. The ISIQ_IGItOISIM_FULFILL_USER_EVENTS environment variable is set to true in ISIQ’s connect-stack.yml file. By default, this environment variable is set to false. For more information about the connect-stack.yml file, see “Configuring the YAML Files” in the [ISIQ Deployment Guide](#).
2. IGI’s EVENT_OUT_USER table exists. For more information, see “Enable the EVENT_OUT_USER table” in “Appendix A: IGI Customizations for ISIQ”.

Updating ERC Status in the IGI OUT Events Queue

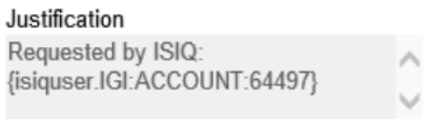
After you subscribe ISIM to IGI, you will notice that the “ERC Status” column in the IGI OUT events queue shows “Unprocessed” for events sent to the ISIM that you configured, even if the events completed. To remedy this situation, check for the creation of a topic with the following name:

<subscriberID>.<isim-name>.ISIM_database.EventCompletion

Once this topic exists, go to the IGI product dashboard in the ISIQ UI and click the “Update subscription” button (for more information, see “Appendix D: Updating Subscriptions”). This action detects and reads the new EventCompletion topic. As a consequence, the “ERC Status” column in the IGI OUT events tab is updated with the result of ISIM operations. The column no longer shows “Unprocessed”.

Interpreting the Justification Field

If you configured the Justification field in the ISIM user interface, here is how to interpret the ISIQ-supplied justification data when an IGI event is sent to ISIM. For example, if you see this field:



Justification
Requested by ISIQ:
{isiquser.IGI:ACCOUNT:64497}

1. Requested by ISIQ – indicates that ISIQ initiated the request.
2. isiquser.IGI – lists the product name where the request originated, so that ISIQ knows which target to update with the result from ISIM.
3. ACCOUNT – indicates whether the event came from IGI’s USER_EVENT_ERC table (labeled “ACCOUNT”) or the EVENT_OUT_USER table (labeled “USER”).
4. 64497 – specifies the request ID in the IGI OUT events table. This ID is the row number that will be updated with the result from ISIM.

Creating Custom Persons in ISIM

If you subscribe ISIM to IGI, any IGI “Create User” events will result in default Person types being created in ISIM. If you want the events to create custom Person types in ISIM, you must modify txdef.json.

Specifically, find the “EVENT_OUT_USER” mapping rule under “in”: “igi”, “out”: “isim”, and add the name of your custom profile to the “txe” line as illustrated here:

```
"txe": "{userid: .obj_code, operation: .operation, eventkey: .eventkey, ou: .ou, givenname: .name, dn: .dn, sn: .surname, user_erc: .user_erc, profile: \"CustomPerson\"}"
```

The modification consists of passing in a “profile” attribute along with the name of the profile you want the ISIM user created as. Be sure to include in the “txe” line any other required attributes for the custom Person object class that you’re using.

Ignoring IGI Account Events from a Specific User

If there are IGI account events which originate with a specific IGI user ID, and that are not needed by ISIM, you can tell ISIQ to ignore the events. To leverage the feature, perform these steps:

(1) Add the `ISIQ_IGItoISIM_IGNORE_ACCOUNT_EVENT_FROM_IGIUSER` environment variable to your `connect-stack.yml` file. The value of the environment variable has the following format:

```
<isimProductName>;;<igiUserId>
```

The `<isimProductName>` is the name of the ISIM product you configured in ISIQ. The name is case-sensitive. Here's an example of the environment variable:

```
ISIQ_IGItoISIM_IGNORE_ACCOUNT_EVENT_FROM_IGIUSER=isim10DC;;iq_admin
```

This setting indicates that for IGI-to-ISIM integration, ISIQ will ignore account-related events submitted by IGI user "iq_admin" that are destined for the ISIM system "isim10DC", where "isim10DC" is the ISIM product name you configured in ISIQ.

(2) Ensure the "[cod_operation: .cod_operation](#)" mapping rule is present in the "in": "igi", "out": "isim" section of `txdef.json`. The `cod_operation` attribute is essential because it contains the IGI user ID of the event requester. The default version of `txdef.json` already has the required mapping rule in the `USER_EVENT_ERC` topic transformation. You just need to verify that if you've modified `txdef.json`, your modified copy also includes this mapping rule.

Supported Operations on ISIM and IGI Entities

The following table lists ISIM entities, their corresponding IGI entities, and the operations that ISIQ supports from ISIM-to-IGI and from IGI-to-ISIM:

ISIM Entity	IGI Entity	ISIM-to-IGI Operations	IGI-to-ISIM Operations
Organization, Organizational Unit, Admin Domain, Business Partner Unit	Organizational Unit	Add, Modify, Delete	None
Organizational Role	External Role	Add, Modify, Delete	Add role to a person, Remove role from a person
Group	Permission	Add, Modify, Delete	Add group to an account, Remove group from an account
ISIM System Role (Group in ITIM Service)	Permission	Add, Modify, Delete	Add group to an account, Remove group from an account
Service	Application, Events Marker, Account Configuration	Add, Modify, Delete	None
Person	User	Add, Modify, Delete (see Note 1)	Add

Account	Account	Add, Suspend, Restore, Modify, Delete (see Note 2)	Add, Suspend, Restore, Delete
---------	---------	---	----------------------------------

Notes:

1. When an ISIM person is loaded into IGI, if that person has roles, the corresponding external roles will be assigned to the user in IGI. If a role is added to or removed from an ISIM person, a corresponding external role will be added to or removed from the IGI user.
2. When an ISIM account is loaded into IGI, if the account has groups assigned, the corresponding permissions will be assigned to the IGI user. If a group is added to or removed from an ISIM account, a corresponding permission will be added to or removed from the IGI account.

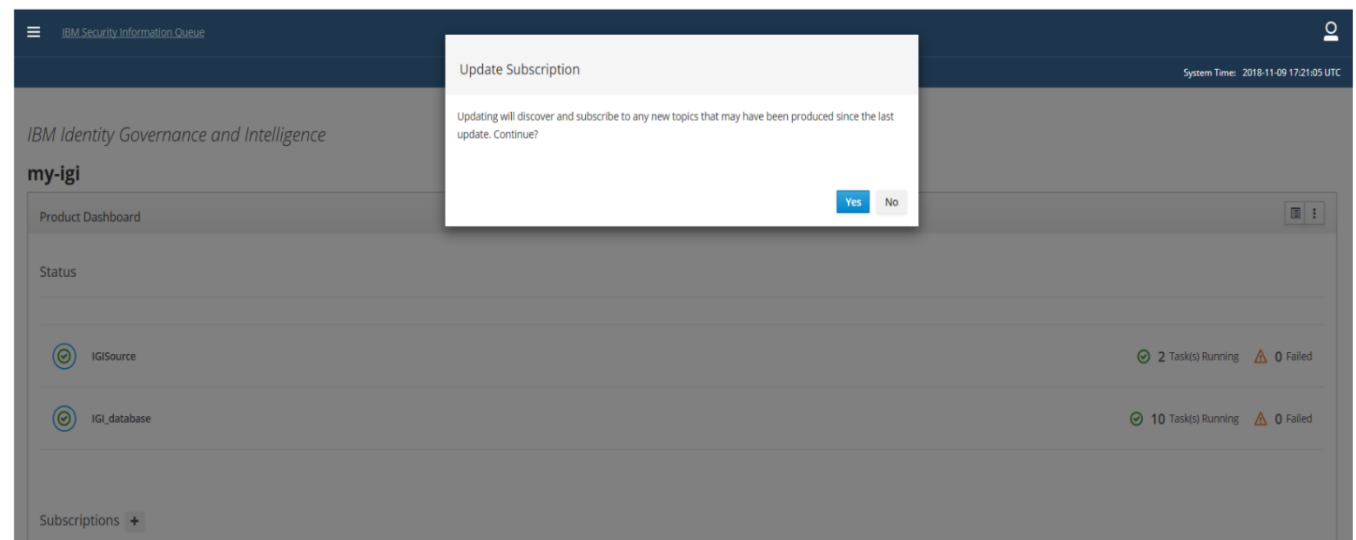
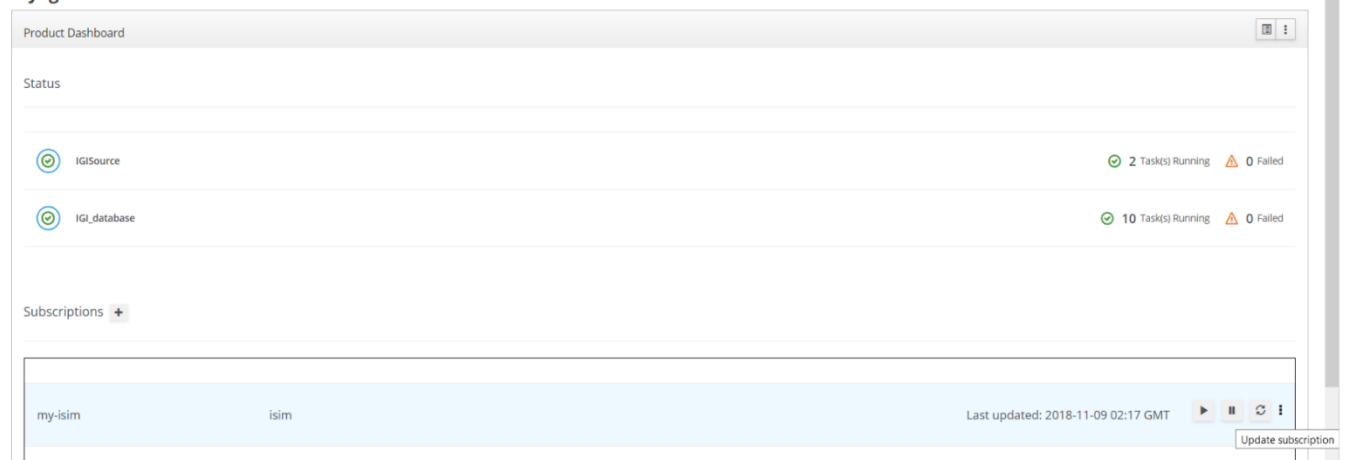
Appendix D: Updating Subscriptions

At the time of subscription creation, ISIQ transforms data from available source topics according to the transformation rules defined in the template file, txdef.json. However, there might be cases in which new source topics get added, because of new source data, after subscription creation. These new topics are not automatically detected by the existing subscription. For example, if an ISIM user adds a new SAP adapter service profile after the IGI-to-ISIM subscription was created, the subscription won't know about the SAP service entity topics.

To address this type of scenario, there is an “Update subscription” menu option on the ISIQ product dashboard page.

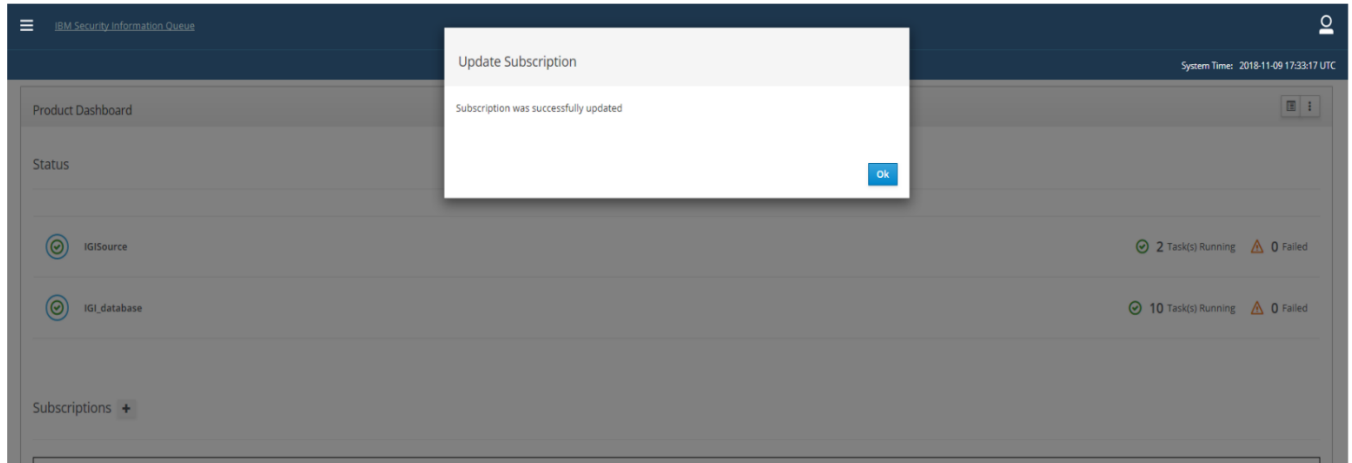
IBM Identity Governance and Intelligence

my-igi



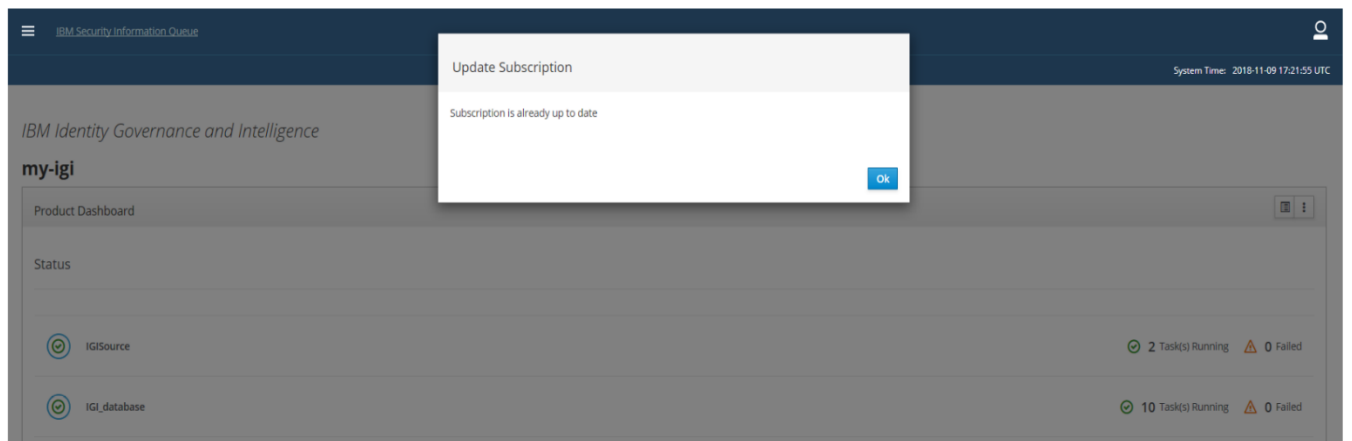
Selecting “Yes” causes the subscription to

1. Reevaluate the list of available topics and apply defined transformation rules, and
2. Subscribe to any new data that has been produced since subscription creation.



In our example, the data from the new SAP service is transformed and consumed by the subscribing product, effectively updating the subscription's transformation rules.

There's no harm in invoking the "Update subscription" menu option if your subscription is currently up to date. You will see this informational message box:



Here's a related scenario to be aware of: When you subscribe an IBM Security Identity Manager product to IGI, the IGISource connector only creates topics for events it receives *after* the connector started, which means the IGI USER_EVENT_ERC topic won't exist until at least one IGI account has been changed. If hours or days elapse without an IGI account change, your ISIM-to-IGI subscription won't know that it must also monitor the USER_EVENT_ERC topic.

You can check for this scenario by running the topicList.sh script provided in the /util directory, as extracted from the ISIQ starter kit .zip file. You can also look at the "Topic List" UI described earlier in "Connectors and Topics".

In the list of topics, search for entries belonging to your IGISource connector, such as <subscriberID>.my-igi.IGISource.EVENT_OUT_USER and <subscriberID>.my-igi.IGISource.USER_EVENT_ERC. Once you confirm those topics exist, go to the ISIQ dashboard for the configured IBM Security Identity Manager product and click the "Update subscription" menu option for your IGI source. The subscription re-reads the topic list to ensure all the data is seen. Any delays in updating the subscription will not cause loss of data. It simply means that user events won't flow from IGI to ISIM until the update action is taken.

Appendix E: Custom ISIM-to-IGI Transformations

This Appendix describes the options you have for customizing ISIM-to-IGI integration. The most common method is to update the txdef.json file. For the basic syntax and format of txdef.json, see the “Custom Transformations” section earlier in this guide. **Note:** Before updating txdef.json, always be sure to back up the file.

There are several use cases in which you might want to implement customized ISIM-to-IGI integration. For example, you have extended Person attributes that you want to map to IGI attributes in a particular way, or you don’t want ISIM persons to be synchronized into IGI with their default ISIM Organizational Unit (OU) names. Instead, you prefer to assign them to a specially created IGI OU. This Appendix explains how to implement those types of customizations.

In the default txdef.json, there are two sections for user transformation as shown here:

```
{
  "in": "Person",
  "out": "USER_ERC",
  "txe": "{pm_code: .uid[0], ou: .erparent, given_name: .givenname[0], surname: .sn[0], email: .mail[0],
address: .postaladdress[0], erroles: .erroles, eralias: .eralises, isiq_user_dn_column: \"ATTR11\",
isiq_acct_owner_column: \"ATTR8\"}"
},
{
  "in": "BPPerson",
  "out": "USER_ERC",
  "txe": "{pm_code : .ercustomdisplay[0], ou: .erparent, given_name: (.cn[0]|split(\" \")|. [0]), surname:
.ercustomdisplay[0], email: .mail[0], erroles: .erroles, isiq_user_dn_column: \"ATTR11\",
isiq_acct_owner_column: \"ATTR8\"}"
},
```

As you can tell, the transformation for users must have USER_ERC as the “out” topic. Person and BPPerson are object profile names for ISIM users. After you configure an IBM Security Identity Manager product in ISIQ, you should have “Person” and “BPPerson” topics that hold person and business partner person data. For example, if you list all of your ISIQ topics, you might see:

```
myuserid.myISIM.directory.Person
myuserid.myISIM.directory.BPPerson
```

By default, the only persons that ISIQ transforms are Person and BPPerson objects. For more information about supporting custom Person and BPPerson profiles, see “How to Support Custom Persons and BPPersons” later in this Appendix.

An ISIM object profile contains information about various types of entities. The transformation section (“txe”) of txdef.json specifies the list of transformations for each attribute of an ISIM entity. The format of the transformation is:

<IGI attribute name>:<JQ filter for transforming ISIM attribute value>

The IGI attribute name cannot be changed since it’s part of a fixed set of IGI attributes. You can only customize the JQ filter for an ISIM attribute value. In other words, you can only customize which ISIM attribute to map to an IGI attribute, and how to perform the transformation between the two attributes. This same rule applies to all entity types.

In the txdef.json file, there are some IGI attribute names prefixed with er, iq, or isiq, such as erroles, iq_group_name, and isiq_user_dn_column. They are not IGI attribute names. They are used internally by ISIQ for a special purpose. In general, they must not be changed except for the ones that have ATTRnn as the transformation filter. Another exception would be if you are using an adapter in which you store the unique group name in the ergroupid attribute rather than in the ergroupname attribute. In that case, change the "txe" line to set groupname to .iq_groupid instead of to .iq_groupname.

How to Customize Transformations for Accounts

By default, a limited number of ISIM source attributes are transformed into IGI account attributes. Here is an excerpt from txdef.json that illustrates the default transformations:

```
{
  "in": "ITIMAccount",
  "out": "ACCOUNT",
  "txe": "{name: .eruid, service: .erservice, groups: .erroles, status: .eraccountstatus, owner: .owner[0], isiq_acct_owner_column: \"ATTR8\"}",
},
{
  "in": [
    "ADAccount",
    "AzureAccount",
    ...
    ...
    "ZDB2AdapterAccount"
  ],
  "out": "ACCOUNT",
  "txe": "{name: .eruid, service: .erservice, groups: .iq_groups, status: .eraccountstatus, owner: .owner[0], isiq_acct_owner_column: \"ATTR8\"}",
},
```

As you see, it's a small set of IGI account attributes, and you cannot modify the IGI attribute names. There are two means of customizing this set: (1) change the ATTR8 mapping; and (2) assign values for the "NAME", "SURNAME", "DISPLAY_NAME", and "EMAIL" columns in the IGI EVENT_TARGET table.

(1) In the "Update ACCOUNT_CREATE rule flow" section of Appendix A, the EVENT_TARGET.ATTR8 column was leveraged to set the account owner DN so that the account could be matched with the account owner. If ATTR8 is already being used for another purpose in the EVENT_TARGET table, then you need to do the following:

- Decide which ATTRnn to use in the table instead of ATTR8.
- Update "Create Account - DN Matching Rule" to use ATTRnn.
- Replace all occurrences of ATTR8 with ATTRnn in txdef.json.
- Restart the ISIQ stacks.
- Update the IGI-to-ISIM subscription.

(2) The "name", "surname", "display_name", and "email" attributes are a standard part of an IGI account. However, in the default txdef.json, none of the four is present (with the exception of "name"; see the Note: below) because not all ISIM accounts contain this data, and when they do, it's often in a profile-specific source attribute that is not applicable for all customers.

To populate these attributes in your IGI accounts, you need to modify txdef.json. We recommend you separate out into its own stanza the account profile that corresponds to your ISIM account type. For example, if you are loading ISIM AD accounts into IGI, move “ADAccount” from the list of account profiles into its own stanza. The new ADAccount stanza in txdef.json might look like this:

```
{
  "in": [
    "ADAccount"
  ],
  "out": "ACCOUNT",
  "txe": "{isim_attrs: ., name: .eruid, service: .erservice, groups: .iq_groups, status: .eraccountstatus,
owner: .owner[0], isiq_acct_owner_column: \"ATTR8\", display_name: .eraddisplayname, givenname:
.givenname[0], surname: .sn[0], email: .eradupn}"
},
```

This example includes all four additional attributes, but you can specify fewer if you only require a subset. After saving the new stanza in txdef.json, restart the ISIQ stacks and update your subscription to implement the change.

Note: One possible area of confusion is the handling of the “name” account attribute. It was previously mapped to “.eruid” at the start of the “txe” list, and so ISIQ must map a different attribute to fill in the NAME column in the IGI EVENT_TARGET table. When processing accounts, the IGI sink connector uses “givenname” as a substitute for “name”. In the ADAccount example, “givenname” is mapped to “.givenname[0]” but you can choose any source attribute. The IGI sink connector sees “givenname” in the transformed record, and inserts the associated source attribute value into the NAME column.

How to Customize Transformations for Users

ISIQ leverages the USER_ERC table to create, modify or delete IGI users. That’s why USER_ERC is defined as the “out” topic for ISIM “Person” and “BPPerson” entities in txdef.json.

ATTR1, ATTR2, and up through ATTR15 are extended attribute columns in the USER_ERC table. Some of these fifteen extended attributes are already in use, but others are available to set additional attribute values for IGI users.

Follow these steps to use ATTRnn to bring over additional ISIM person attributes to an IGI user:

1. Find out which ATTRnn extended attributes are available. The extended attributes that are not shown on IGI’s UserErc Attribute Mapping page are eligible for your use (for more information about the IGI UI, click the Help link from this page).

IBM Security Identity Governance and Intelligence Access Governance Core Ideas / admin Help Logout

Manage Configure Monitor Tools Settings

Core Configurations Configure Password Service

General User Virtual Attributes Internal Events

Repository Details Attribute Mapping

Actions

Enabled	Name
	S_User
	Swim_User...
	UserErc

Save Cancel Actions

	Visible	Position	Required	Key	Name	Label
☐	☒	↑ ↓	☐		ACCOUNT_EXPIRY_DATE	ACCOUNT_EXPIRY...
☐	☒	↑ ↓	☐		ADDRESS	_ADDRESS
☐	☒	↑ ↓	☐		ATTR1	Manager
☐	☒	↑ ↓	☐		ATTR10	ATTR10
☐	☒	↑ ↓	☐		ATTR11	_DN
☐	☒	↑ ↓	☐		ATTR2	Is Dep. Manager
☐	☒	↑ ↓	☐		ATTR3	Education
☐	☒	↑ ↓	☐		ATTR4	Cod Area
☐	☒	↑ ↓	☐		ATTR6	Cod Subarea
☐	☒	↑ ↓	☐		ATTR7	Cod User
☐	☒	↑ ↓	☐		BIRTH_COUNTRY	_BIRTH_COUNTRY

2. As an example, suppose you decide to use ATTR13 to set ISIM's postaladdress attribute value. You need to add ATTR13 with your designated label on the UserErc Attribute Mapping page.

3. Update the txdef.json file. ATTR13:postaladdress[0] must be added to the "txe" section.

```
{
  "in": "Person",
  "out": "USER_ERC",
  "txe": "{pm_code: .uid[0], ou: .erparent, given_name: .givenname[0], surname: .sn[0], email:
.mail[0], address: .postaladdress[0], erroles: .erroles, eralias: .eralias, isiq_user_dn_column:
\"ATTR11\", isiq_acct_owner_column: \"ATTR8\", ATTR13:postaladdress[0]}"
},
```

4. Restart the ISIQ stacks (note: for more information about this step and the next step, see the relevant sections at the end of this Appendix).

5. Update the IGI-to-ISIM subscription.

In the Appendix A section, "Update the USER_ERC attribute mapping to set the user DN", USER_ERC.ATTR11 is mapped to PERSON.DN. But let's suppose that ATTR11 had been allocated for some other purpose. In that circumstance, you could switch to another available ATTRnn:

1. Update the USER_ERC Attribute Mapping page with the other ATTRnn.
2. Replace all occurrences of ATTR11 with ATTRnn in the txdef.json file.
3. Restart the ISIQ stacks.
4. Update the IGI-to-ISIM subscription.

In addition to the ATTR1-ATTR15 extended attributes in USER_ERC, any other attribute in the table can be mapped and supported by ISIQ. Consider the case in which you want to set the value of USER_TYPE in the USER_ERC table to an available attribute in the PERSON table. Since the PERSON table offers

ATTR1-ATTR5 as its extended attributes, you decide to use ATTR5. To accomplish the mapping, perform these steps:

1. Verify that the attributes you want to map are available and map them to the respective fields in the Attribute Mapping UI. Note: In this screen capture, the underscore character that prefixes the _ATTR5 Label denotes a reference to the PERSON table. Without the underscore prefix, the Label specified there would simply be mapped to the IGI Name.

The screenshot shows the IBM Security Identity Governance and Intelligence Access Governance Core interface. The top navigation bar includes 'Manage', 'Configure', 'Monitor', 'Tools', and 'Settings'. The 'Configure' tab is active, and the 'User Virtual Attributes' sub-tab is selected. The 'Repository' section on the left lists three attributes: 'S_User', 'Swim_User...', and 'UserErc'. The 'Details' section on the right shows the 'Attribute Mapping' configuration. A table lists attributes and their mappings to labels.

Visible	Position	Required	Key	Name	Label
☐		☐		CUSTOM_ATTR1	Custom_Attr1
☒		☐		USER_TYPE	_ATTR5
☐		☐		ATTR11	_DN
☐		☐		ATTR10	ATTR10
☐		☐		PHONE_NUMBER	_PHONE_NUMBER
☐		☐		COUNTRY	_COUNTRY
☐		☐		ZIPCODE	_ZIPCODE
☐		☐		NATION	NATION

2. Update the txdef.json file as shown here, with "user_type" added to the "txe" section:

```
{
  "in": "Person",
  "out": "USER_ERC",
  "txe": "{pm_code: .uid[0], ou: .erparent, given_name: .givenname[0], surname: .sn[0], email:
.mail[0], address: .postaladdress[0], erroles: .erroles, eralias: .eralias, isiq_user_dn_column:
\"ATTR11\", isiq_acct_owner_column: \"ATTR8\", user_type: \"Default Person\"}"
},
{
  "in": "BPPerson",
  "out": "USER_ERC",
  "txe": "{pm_code: .uid[0], ou: .erparent, given_name: .givenname[0], surname: .sn[0], email:
.mail[0], address: .postaladdress[0], erroles: .erroles, eralias: .eralias, isiq_user_dn_column:
\"ATTR11\", isiq_acct_owner_column: \"ATTR8\", user_type: \"Business Person\"}"
},
}
```

3. Restart the ISIQ stacks.
4. Update the IGI-to-ISIM subscription.

How to Support Custom Persons and BPPersons

To allow ISIQ to transfer ISIM custom Person and BPPerson entities, follow these steps:

Step 1: Confirm your entities have been defined, and objects of that type have been created in ISIM. This is typically already the case, but if not, define the custom Person or BPPerson entity and create at least one entry of that type before moving to Step 2.

Step 2: Decide how best to update txdef.json. If you have a custom Person class, and the default Person mapping rules are applicable to it, then you simply need to add your profile name to the "in" line for the existing Person entry.

If you don't know the profile name of your custom Person class, you can run topicList.sh in the /util directory to display all topics (you can also view the "Topics" link on the System Health dashboard). In the topics that ISIQ generates for ISIM, the profile name gets stored in the last level of the topic name. For example, if your custom Person class is named "IQPerson", you would see a topic such as this:

myuserid.myISIM.directory.IQPerson

Using the IQPerson example, make these changes to txdef.json:

```
"in": "Person",  
"out": "USER_ERC",  
to:  
"in": [ "Person", "IQPerson" ],  
"out": "USER_ERC",
```

Notice how the "in" line was converted from a JSON string to a JSON array because there is now more than one type of Person. If you have other custom Person classes that use the same transformation rule, add them as other comma-separated entries in this array.

However, if your custom Person class requires a different mapping relationship between ISIM and IGI attributes, you need to create a new entry in the txdef.json file. Copy the lines for Person, paste them after the end of the Person entry, then customize as needed. The full entry might look like this:

```
}, // End of the Person entry  
{  
  "in": "IQPerson",  
  "out": "USER_ERC",  
  "txe": "{pm_code: .employeeNumber, ou: .erparent, given_name: .givenname[0], surname: .sn[0],  
email: .companymail[0], address: .postaladdress[0], erroles: .erroles, eralias: .eraliases,  
isig_user_dn_column: \"ATTR11\", isig_acct_owner_column: \"ATTR8\"}"  
},
```

The full entry includes the open brace "{" through the closing brace and comma "},". In this example, the line "}, // End of the Person entry" was included to illustrate how to append your custom Person entry after the default Person entry.

The process operates the same when working with a custom BPPerson. Just use the default BPPerson entry in txdef.json as the starting point for your modifications.

Step 3: Once the custom person data exists in ISIM and the txdef.json file has been updated with the appropriate mapping, ISIQ must discover this new custom person data, which requires

- Restarting ISIQ. For more information about docker stop and start commands, see the [ISIQ Deployment Guide](#) sections, "Removing the ISIQ stacks" and "Deploying the ISIQ stacks".

- Updating the IGI-to-ISIM subscription on the IGI product page to allow the new data to get sent to ISIQ topics and loaded into IGI.

How to Support a Custom Service

In ISIM, if you have a custom service type, you need to update the txdef.json file so that ISIQ can load the services, groups, and accounts for that custom service type into IGI. Here are the ISIQ steps for supporting a custom service:

1. Find the object profile name for the service profile, group profile, and account profile.
2. Update txdef.json with the new profile names.
3. Restart the ISIQ stacks.
4. Update the IGI-to-ISIM subscription.

The following example uses the sample command-line service type to demonstrate the customization process. The example was developed by using the Command Line (CLIX) Adapter (refer to <https://www.ibm.com/support/pages/verify-adapters-product-documentation-pdfs>).

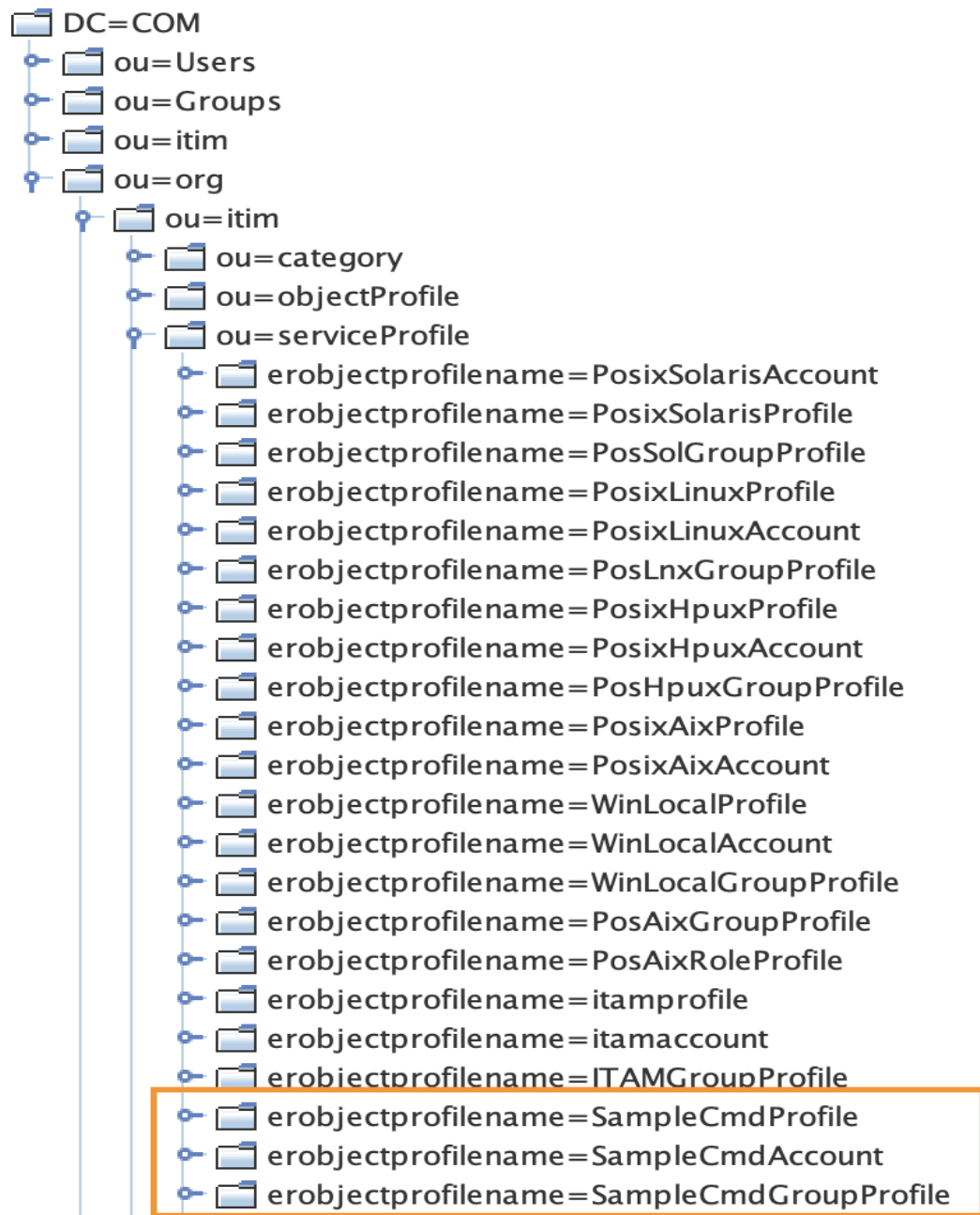
Find Object Profile Name for Service, Group, and Account

In ISIM, an object profile represents different types of ISIM entities. There can be different profiles in the same entity type category. For example, the POSIX AIX service profile and POSIX Linux service profile are two different service profiles in the Service category. The POSIX AIX account profile and POSIX Linux account profile are two different profiles in the Account category.

After an adapter profile JAR file is imported into ISIM, the adapter's service profile, group profile, and account profile are created in the LDAP server. A profile has information about its respective service, group, or account. It also has information about the relationship with the other two types.

Every adapter has a service profile and account profile. Some adapters do not have a group profile if groups are not managed by the adapter. Some adapters (such as the POSIX AIX adapter) have multiple group profiles if multiple group types are managed by the adapter.

If you are familiar with service profiles, you can find the service, group, and account profile names from the LDAP directory server as shown here:



For this sample CLlx service type, **SampleCmdProfile** is the service profile name, **SampleCmdAccount** is the account profile name, and **SampleCmdGroupProfile** is the group profile name.

If you are not familiar with ISIM service profiles, follow these steps to find the service, group, and account profile names for your custom service type:

1. From the ISIM Administration Console, log in as an administrator user such as ITIM Manager, select Configure System, and select Manage Service Types.
2. Find the custom service type and select the name.

ITIM Manager: My Work

- Home
- Change Passwords
- Manage Roles
- Manage Organization Structure
- Manage Users
- Manage Services
- Manage Orphan Accounts
- Manage Groups
- Manage Policies
- Design Workflows
- Set System Security
- Reports
- Configure System**
 - Manage Service Types**
 - Manage Access Types
 - Manage Ownership Types
 - Global Adoption Policies
 - Workflow Notification Properties
 - Post Office
 - Design Forms
 - Manage Entities

Manage Service Types

Configure System > Manage Service Types

You can add, import, change, or delete service types. Select the service type in the table, and then click the

Create Import... Change Delete Refresh

Select	Service Type	Description
☐	IBM Security Access Manager Profile	IBM Security Access Manager service profile
☐	ITIM	Default Service Type used for ITIM accounts
☐	LDAP profile	LDAP service profile
☐	POSIX AIX profile	AIX accounts service profile
☐	POSIX HP-UX profile	HP-UX accounts service profile
☐	POSIX Linux profile	Linux accounts service profile
☐	POSIX Solaris profile	Solaris accounts service profile
☐	RACF SSL Profile	RACF Service Profile 7.1.28
☐	Sample Command-line profile	Sample Command-line profile
☐	winlocalProfile	No description provided

Page 1 of 1 Total: 10 Displayed: 10 Selected: 0

3. From the service type form, select the *Service tab, and write down the value for LDAP class, **erSampleCmdService**.

Manage Service Types

Configure System > Manage Service Types > Service

Type the service schema information. You must click Search to specify the super class. When you are done specifying information on each of the tabs, click OK.

*LDAP class

erSampleCmdService

4. From the service type form, select the *Account tab, and write down the value for LDAP Class, **erSampleCmdAccount**.

Manage Service Types

Configure System > Manage Service Types > Account

Type the account schema information. You must click Search to specify the super class. When you are done specifying information on each of the tabs, click OK.

*LDAP class

erSampleCmdAccount

5. From the service type form, select the Group tab, select the group name, write down the LDAP class value, **ersamplecmdgroups**.

Manage Service Types			
- General - Service - Account Group - Miscellaneous	Configure System > Manage Service Types > Group You can define group schema associated with this service type. A Group schema should be supported by the adapter for this service type. <table border="1"> <thead> <tr> <th>Group Name</th> </tr> </thead> <tbody> <tr> <td>erSampleCmdGroups</td> </tr> </tbody> </table> Page 1 of 1 Total: 1 Displayed: 1 Selected: 0	Group Name	erSampleCmdGroups
Group Name			
erSampleCmdGroups			

Manage Service Types	
Configure System > Change Group Specify the service group schema information, and click OK	
*LDAP class ersamplecmdgroups Object identifier (OID) erSampleCmdGroups-OID	

If your custom service has multiple group types, then you need to find out the LDAP class value for each group type. As an example, if you look at the POSIX AIX profile (service type), it has two group types: one for AIX group and another one for AIX role.

Manage Service Types				
- General - Service - Account Group - Miscellaneous	Configure System > Manage Service Types > Group You can define group schema associated with this service type. A Group schema should be supported by the adapter for this service type. <table border="1"> <thead> <tr> <th>Group Name</th> </tr> </thead> <tbody> <tr> <td>erPosixAixGroup</td> </tr> <tr> <td>erPosixAixRole</td> </tr> </tbody> </table> Page 1 of 1 Total: 2 Displayed: 2 Selected: 0	Group Name	erPosixAixGroup	erPosixAixRole
Group Name				
erPosixAixGroup				
erPosixAixRole				

6. From the LDAP directory server, search the service profile which has **erSampleCmdService** for its **ercustomclass** value, as shown in the next screen capture. The Search DN is set to the base DN where all service profiles can be located. The "ou=org,DC=COM" substring represents the tenant DN. Your tenant DN might be different than what is shown in this example:

Search

Search DN: ou=serviceProfile, ou=itim, ou=org, DC=COM

Filter: (ercustomclass=erSampleCmdService)

Attributes:

Search scope: ☐ One level ☒ Sub-tree level

dn
erobjectprofilename=SampleCmdProfile, ou=serviceProfile, ou=itim, ou=org, DC=COM

Matched 1 entry.

Search Export Cancel

The object profile name (erobjectprofilename) value for this service type is **SampleCmdProfile**. The screen capture shows erobjectprofilename as being part of the DN.

- Find the account profile name using the account LDAP class, **erSampleCmdAccount**. The account profile name is **SampleCmdAccount**.

Search

Search DN: ou=serviceProfile, ou=itim, ou=org, DC=COM

Filter: (ercustomclass=erSampleCmdAccount)

Attributes:

Search scope: ☐ One level ☒ Sub-tree level

dn
erobjectprofilename=SampleCmdAccount, ou=serviceProfile, ou=itim, ou=org, DC=COM

Matched 1 entry.

Search Export Cancel

- Find the group profile name using the group LDAP class, **ersamplecmdgroups**. The group profile name is **SampleCmdGroupProfile**.

Search

Search DN:

Filter:

Attributes:

Search scope: ☐ One level ☒ Sub-tree level

dn
erobjectprofilename=SampleCmdGroupProfile, ou=serviceProfile, ou=itim, ou=org, DC=COM

Matched 1 entry.

You have now found the service, group, and account profile names as shown here:

- Service profile name: **SampleCmdProfile**
- Group profile name: **SampleCmdGroupProfile**
If you have multiple group profiles (types) for your custom service, you need to find object profile names for all group profiles.
- Account profile name: **SampleCmdAccount**

Update txdef.json with the new profile names

Save a copy of <isiqStarterKit>/cfg/connect/txdef.json before you modify the file. Update txdef.json with the service, group, and account profile names:

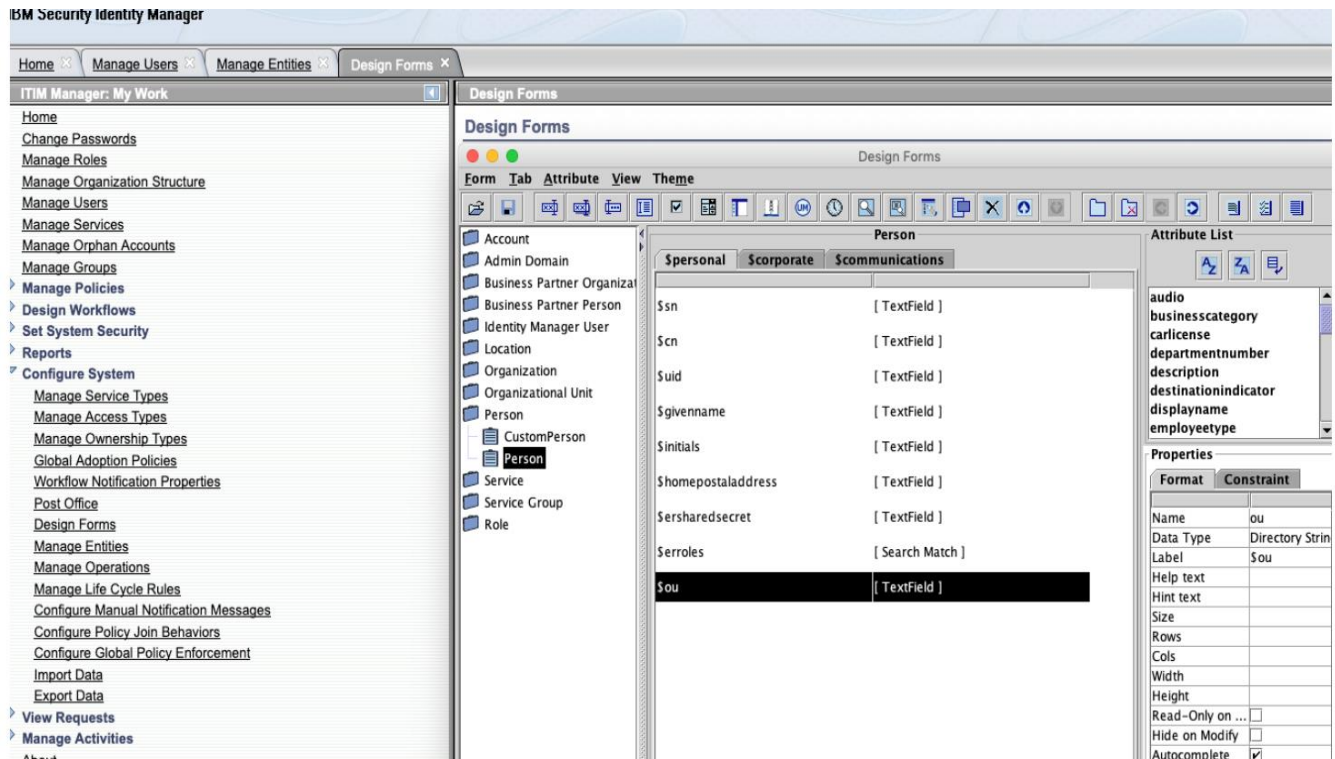
1. In the ISIM-to-IGI section, find the section for service entity mapping and add the service profile name to the input ("in") topic list. The ISIM-to-IGI section is the section where it has "in": "isim" and "out": "igi". The service entity-mapping section is where it has "out": "SERVICE". You may see multiple sections where it has SERVICE as output. The ones with multiple service profiles are the ones using the default transformation. If the default transformation for the service is not what you want for your custom service, you might need to have a separate section for your service profile. For this sample CLI service type, add the service profile name **SampleCmdProfile** at the end of the service profile list for the default transformation.

```

{
  "in": "isim",
  "out": "igi",
  "smts": [
    {
      ..
    },
    {
      ..
    },
    {
      ..
    },
    {
      ..
    },
    {
      "in": [
        "ITIMProfile",
        "ADprofile",
        "AzureProfile",
        ..
        ..
        "WinLocalProfile",
        "WinlocalProfile",
        "ZDB2AdapterProfile",
        "SampleCmdProfile"
      ],
      "out": "SERVICE",
      "txe": "{name: .erservicename, description: .description[0], objectclas
    },
  ],
}

```

2. In the ISIM-to-IGI section, find the section for the group entity mapping and add the group profile name. If the default transformation for the group is not what you want for your custom group, you might need to have a separate section for your group profile. For this sample CLI service, add the group profile **SampleCmdGroupProfile** at the end of the list for the default transformation. If you have multiple group types in your custom service type, you need to add each one of them to the list.



As an attribute of organizationalPerson, “ou” will be available to Person, BPPerson, custom Person, and custom BPPerson objects.

2. Update ISIQ's connect-stack.yml file by inserting the following environment variables under “services:”
“connect:”
“environment:”:

ISIQ_SKIP_OU_SYNC_ISIM_PERSON_OU_ATTR=jsong0104.btisim6:ou

Specifies which person attribute will contain the OU code in IGI. In our example, the attribute was called “ou”. The value of this environment variable must start with an ISIM key prefix in the format, “<subscriberID>.<isimProductName>:”.

ISIQ_SKIP_OU_SYNC_ISIMtoIGI_DEFAULT_OU_CODE=jsong0104.btisim6:igiDefaultOu

Specifies a default OU in case the ISIM person has an invalid OU.

ISIQ_SKIP_OU_SYNC_IGItoISIM_DEFAULT_OU_DN=jsong0104.btisim6:erglobalid=8354189953611274681,ou=orgChart,erglobalid=00000000000000000000,ou=org,dc=com

Specifies the OU DN in ISIM that will be assigned after a new user is created in IGI.

For these environment variables, you can provide multiple values by using three semicolon characters (;;;) as a delimiter. For example, if you have two IBM Security Identity Manager products, and you want to implement the same “Skip OU Synchronization” behavior in both of them, here’s what you specify:

ISIQ_SKIP_OU_SYNC_ISIM_PERSON_OU_ATTR=mySubscriberID.isim1:ou;;;mySubscriberID.isim2:ou

If you have a two-way subscription between ISIM and IGI, any new IGI users get synced to ISIM into the OU that you specified in the ISIQ_SKIP_OU_SYNC_IGItoISIM_DEFAULT_OU_DN environment variable. When a new person is created in ISIM, its “ou” attribute is set according to that environment variable.

3. Update txdef.json file so that IGI's OU column in the USER_ERC table can be mapped to the ISIM person's "ou" attribute. By default, your txdef.json will contain "ou: .erparent". The following excerpt shows the change that you make (highlighted in black) for "Person". Because you must have a common attribute for all Person types, you need to repeat this change for all Person entities in txdef.json. Since "ou" is a multi-valued attribute, notice that it uses the array format, .ou[0]:

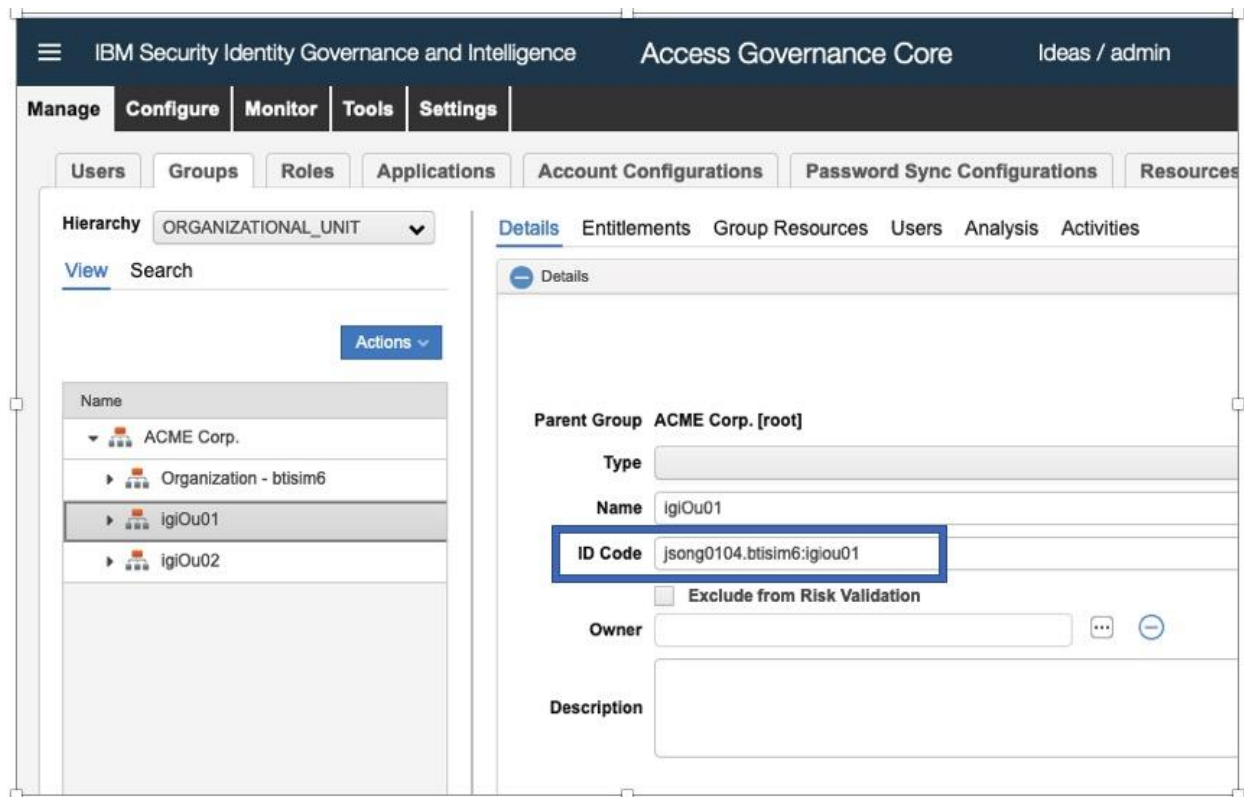
```
{
  "in": ["Person"],
  "out": "USER_ERC",
  "txe": "{pm_code: .uid[0], ou: .ou[0], given_name: .givenname[0], surname: .sn[0], email:
    .mail[0], address: .postaladdress[0], erroles: .erroles, eralias: .eralias, isiq_user_dn_column:
    \ATTR11\", isiq_acct_owner_column: \ATTR8\", ATTR12: .employeenumber}"
},
```

To ensure that no IBM Security Identity Manager OUs get loaded into IGI, remove all txdef.json sections for the different types of OUs, specifically, any section whose "out" value is OU_ERC, as shown here:

```
{
  "in": "Organization",
  "out": "OU_ERC",
  "txe": "{name: .o[0], description: .description[0]}"
},
{
  "in": [
    "AdminDomain",
    "BPOrganization",
    "Location",
    "OrganizationalUnit"
  ],
  "out": "OU_ERC",
  "txe": "{name: (.ou[0]//.l[0]), description: .description[0], erparent: .erparent, ersupervisor:
    (.administrator[0]//.ersponsor[0]//.ersupervisor)}"
},
```

Refer to the "Custom Transformations" section in this User's Guide for txdef.json syntax.

4. On IGI's Access Governance Core -> Manage -> Groups page, make sure that the OU's "ID Code" matches the <subscriberID>.<isimProductName>.<igioucode> format.



The ID Code must be lowercase. If you specify mixed case such as “Jsong0104.Btisim6:igiOu01”, ISIQ cannot match it to the lowercase value set from ISIM's “ou” attribute.

5. Once you’ve defined a person attribute to hold the IGI OU code, you need to make sure each ISIM person has the OU code, which is “igiou01” in the following screen capture:

Manage Users > Change User > Personal Information	
+Personal Information	Type the appropriate information for the user. When you are done specifying it, Click Submit Now to change the user immediately or Schedule Submission to :
Business Information	
Contact Information	
Assignment Attributes	
	+Last name p01 +Full name p01 p01 +Preferred user ID p01 First name p01 Initials aa Home address aaa Shared secret Organizational roles orgRole01 Add orgRole01 Search... Delete Organizational unit name igiou1
Submit Now Schedule Submission Cancel	

6. Implement your “Skip OU Synchronization” changes by performing these actions:

- Remove the ISIQ app stack first and then remove the connect stack.
- Deploy the connect stack and then deploy the app stack.
- Use the “Reprocess” button on the ISIQ product's Subscription page to select the “Person” topic and reload ISIM persons. With the “ou” attribute created, the environment variables specified in connect-stack.yml, and the txdef.json updates, all ISIM persons will be placed in the preferred IGI OU if the persons have an “ou” attribute value that matches IGI’s ID code.

Additional ISIM Customization Options

The <starterKitDir>/cfg/connect/isimConfiguration.json file offers additional options for customizing how ISIM data is read or skipped. Because the file is a JSON array, you can use it to customize multiple ISIM servers. In most cases, only one ISIM server is configured in ISIQ, and thus the default “global” tenant in isimConfiguration.json is sufficient. But if you configured more than one ISIM in ISIQ, you can define each one in isimConfiguration.json with a corresponding “tenant” DN value. When adding a non-global “tenant” definition, be aware of these factors:

- (1) It should be the same DN you specified as “Tenant DN” when you configured the IBM Security Identity Manager product in ISIQ. You must also prepend the DN with the LDAP host, port number, and a ‘/’ delimiter, for example:

“tenant”: “ldapHostA:389/ou=org,dc=com”,

(2) There's no inheritance from the global tenant, in other words, the customizations don't begin with the global settings and then override them with any that you specify for a non-global tenant. ISIQ uses the settings for whichever entry matches. You can either default to the global tenant, or you can specify your own tenant field with a DN corresponding to your IBM Security Identity Manager product and use its settings in place of the global tenant.

Here is a list of the customizations you can implement with `isimConfiguration.json`:

Support Auxiliary Object Classes

When you create custom entities in ISIM, you define them using a combination of structural and auxiliary object classes. ISIQ only knows about attributes in the default ISIM object classes and in the "ercustomclass" object class. If extra ISIM attributes are defined using additional auxiliary object classes, the extra attributes will be rejected by ISIQ as invalid since they belong to unknown classes.

To get around this restriction, the `isimConfiguration.json` file allows you to specify one or more object classes that should be used for a given `ercustomclass`. As an example, note the "auxClasses" field listed at the end of the default `isimConfiguration.json`:

```
"auxClasses": {  
    "OrganizationalUnit": [ "iqExtraOrg" ]  
},
```

These statements define an "iqExtraOrg" auxiliary object class so that ISIQ can successfully load its attributes into Kafka topics.

If you see Data Exception errors in your ISIQ connect service log, such as
 companyname is not a valid field name

And you know you have an attribute called "companyname" defined in an ISIM custom OU, then you need to add the name of that custom OU as an auxiliary object class to the "OrganizationalUnit" section of `isimConfiguration.json`.

Configure LDAP Connection Pool

When you configure an IBM Security Identity Manager product in ISIQ, the most frequent reads are performed against the LDAP directory. To speed data integration, the ISIQ directory source connector starts ten worker threads to run multiple LDAP reads in parallel, and so pooling is helpful to manage the multiple connections.

ISIQ uses the Apache LDAP implementation, which supplies a set of configuration parameters for its connection pool. To learn more about the parameters, select the "GenericObjectPool" class on the Javadoc page for the Apache Commons API:

<https://commons.apache.org/proper/commons-pool/apidocs/index.html>

In `isimConfiguration.json`, the "ldapPool" field under "tenant": "global" lists default values for the connection pool parameters. The defaults are adequate for most purposes, but you are free to override them in "global", or in your own "tenant" definitions. Most of the connection pool parameters can be understood from the Javadoc page. However, there are two that need more explanation:

(1) "whenExhaustedAction": This setting determines what action to take when there are no available connections. The setting can have one of three possible numeric codes:

0 = Immediately fail with an exception.

1 = Wait for `maxWaitForConnectionMillis` to see if a connection becomes available before failing with an exception.

2 = Create a new connection when no more are available.

(2) “searchTimeoutMillis”: This is the time in milliseconds that the directory source connector will wait for an LDAP query to complete. The “searchTimeoutMillis” field is not an Apache LDAP connection pool setting. It was previously a global environment variable in ISIQ’s connect stack, but since it belongs with other LDAP configuration information, it was moved under “ldapPool” where it can be customized at the tenant-level if you choose.

Configure LDAP Search Limit and Lookahead Values

By default, for each query to the LDAP directory, the ISIQ directory source connector attempts to retrieve 1000 objects, known as the search limit. To avoid missing any objects that have the same modify timestamp and that cross a search-limit boundary, the source connector buffers the next 100 objects. This is known as the lookahead value. Setting it to 10-20% of the ldapSearchLimit is usually appropriate.

In most circumstances, these defaults are adequate. However, you might have, for example, an LDAP directory that’s not responding quickly. As a result, the recurring queries for 1000 objects take too long to complete, causing Kafka timeouts and other unwanted side effects.

To address these types of situations, you can customize the search limit and lookahead values used by the directory source connector, as shown here:

```
"ldapSearchLimit": 500,  
"ldapLookAhead": 50,
```

In this example, the defaults are reduced to accommodate a slower-performing directory.

Configure LDAP Poll Delay Interval

By default, the ISIQ directory source connector waits 5 seconds before polling for the next batch of directory modifications. If you are concerned about LDAP system overhead and want to slow the process down, you can increase “pollDelaySeconds” above the default 5 seconds. The maximum allowed value is 43200 or 12 hours. The disadvantage of setting a large pollDelaySeconds value is that ISIQ can no longer obtain timely access to LDAP directory modifications.

As an illustration, let’s suppose you’re diagnosing a high-CPU problem and wish to temporarily reduce the activity of the ISIQ directory source connector. You decide to set the connector to wait 15 minutes between each polling interval. In that case, add the following parameter to isimConfiguration.json:

```
"pollDelaySeconds": 900,
```

Once the high-CPU problem has been diagnosed, delete the “pollDelaySeconds” parameter and allow the default poll delay value to go back into effect.

Map Embedded ISIM Attribute to IGI Attribute

ISIQ attribute mapping generally operates in a direct manner where ISIM_AttributeA gets assigned to IGI_AttributeA. But occasionally there are scenarios where the mapping relationship is not quite as straightforward, where you want to use ISIM_AttributeA—which is embedded within an ISIM object—as a means of looking up ISIM_AttributeB. You then want to assign ISIM_AttributeB to IGI_AttributeA. To handle situations of this type, you can add one or more lookup definitions in the “embeddedAttributes” section of isimConfiguration.json, for example:

```
"embeddedAttributes": {  
  "Person": [ "manager" ]  
},
```

The format is <profilename>:<attribute array to look up>. The first parameter is an ISIM object. The second parameter is treated as an array because you might want to look up multiple attributes. Both the profilename and the attribute array are converted to lowercase, and so it doesn't matter how you specify their case in the definition. The one requirement when using this feature is that the attribute array **MUST** contain a DN. If not, you will encounter an error indicating there was an invalid DN and the data for the attribute lookup won't be filled in.

When would you want to use "embeddedAttributes"? A typical scenario involves the ISIM Manager attribute. It's often assigned as an attribute of a Person object. By using "embeddedAttributes", you could implement the following rule when integrating ISIM data to IGI:

"For every Person object, if it has a Manager attribute value, assign that value to IGI ATTR1."

Besides adding the definition in isimConfiguration.json, you must also code a mapping rule in txdef.json. If we continue the IGI ATTR1 example, the rule might look like this:

ATTR1: ._embedded.manager[0].cn[0]

The format is <IGI ATTR>: ._embedded.<attributeName>[x].<referencedObjectAttribute>

For <attributeName>[x] you can specify something other than array element 0 if it's a multi-value attribute. If it's a single-value attribute, just specify [0]. But you must use the [] syntax because the embedded attribute is treated as an array.

Block Modify Account Events to IGI

After the initial load of ISIM data into IGI, you will notice a repeated set of Modify Account events in IGI's "TARGET inbound - Account events" queue. These events originate with an automated service account. When you configure an IBM Security Identity Manager product in ISIQ, it is common to specify "itim manager" as the "WebServices" user. After you complete product configuration, ISIQ attempts to log in to ISIM every few minutes, using the "itim manager" account, to obtain the status of the web services interface. Each login causes the lastLoginTime attribute for "itim manager" to be updated, which in turn generates a Modify Account event that gets sent to subscribers such as IGI.

If you want ISIQ to block some or all of these Modify Account events from going to IGI, set the "blockMode" and "blockAccounts" fields in isimConfiguration.json. At the end of the default isimConfiguration.json that's shipped with ISIQ, you will see the following name-value pairs:

```
"blockMode": "1",
"blockAccounts": [
    "itim manager"
],
```

In "blockAccounts", you list one or more accounts whose updates will not always trigger a Modify Account event. There are three options for "blockMode", signified by a different numeric code:

0 = All events for this account will be blocked.

1 = The first event after an ISIQ restart will be sent, the rest will be blocked.

2,X = Only process an event for this account if X minutes have elapsed since the last update.

Be sure to place the "blockMode" and "blockAccounts" fields under the tenant—whether it's "global" or a specific tenant—whose Modify Account events you want to suppress.

Reduce ISIM Database Queries

By default, the ISIM database source connector queries for more than a dozen categories of audit events from the AUDIT_EVENT table in the ISIM database. Currently, audit event records are not used when IGI subscribes to ISIM. If you don't have another ISIM subscriber, such as an external application, that

consumes audit event records, you can reduce the overhead of the ISIM DB source connector by shrinking the list of events it queries for.

The "dbTables" field at the end of the default isimConfiguration.json demonstrates how you would specify that only Reconciliation events should be read by the ISIM DB source connector, and all other event categories skipped. If at a later time you need to collect a wider variety of audit events, you can add them back into the "dbTables" list.

If you are in fact consuming many categories of audit events, we recommend you create the ISIM DB indexes listed in the "Tuning Requirements" section of the [ISIQ Deployment Guide](#).

Exclude or Include Services

When a product like IGI subscribes to an ISIM data source, all services are sent. If you want your subscriber to receive only a subset of ISIM services, you must fill in either the "excludeServices" or "includeServices" field in isimConfiguration.json, as illustrated here:

```
"excludeServices": [  
    "4164614613545430761",  
    "2874278754862299112"  
],  
"includeServices": [  
]
```

Enter a comma-separated list of service erglobalids in the desired field. You should provide values for either "excludeServices" or "includeServices", but not both. If you provide both, "includeServices" will get used. The rule of thumb is to specify whichever field allows for a shorter list. If there are a small number of services to exclude, create an "excludeServices" list. On the other hand, if there are only a few services that you want ISIM to send to IGI, create an "includeServices" list. You should also consider what will happen if new services are added in the future. Generally, an "excludeServices" list would be less likely to require changing, assuming that you want to integrate the new services to IGI.

Note: Because "ITIM Service" is a reserved ISIM service, you cannot specify it in an "excludeServices" list. Conversely, if you create an "includeServices" list, it must contain "ITIM Service" in its erglobalid value as "00000000000000000002".

Implement a Naming Attribute Map

The "uid" attribute is important in ISIQ because it serves as a key to achieve parallelism across multiple Kafka partitions. By default, ISIQ uses "uid" for Person and Business Person profiles. If you don't use "uid" for person objects, ISIQ will store all the objects in the same partition, which slows throughput.

To avoid this situation, you can implement a naming attribute map. In the map, you specify the ercustomclass for your person objects and the naming attribute that ISIQ should look at when processing those objects, as shown in this example:

```
"namingAttrMap": [  
    "iqperson": "givenname"  
],
```

Because you're using "givenname" instead of "uid", you map "givenname" to the ercustomclass attribute "iqperson". As a result, ISIQ will leverage "givenname" in the same manner as "uid" and thereby gain the performance benefits of Kafka partitions.

Implement a Profile Topic Map

The ISIM source connector creates a Kafka topic for each non-empty object profile it finds in the LDAP directory. By default, the topic name is assigned based on the object profile name. Due to the limited Latin character set that Kafka supports, it's possible to have ISIM object profiles whose names contain characters disallowed by Kafka.

To circumvent this restriction, you can implement a profile topic map. First, identify any ISIM object profile whose name includes characters other than "a-zA-Z0-9" and "-". Next, find the `ercustomclass` attribute for that object profile. Map the `ercustomclass` attribute to the name of the topic you want the objects to be placed in. Here's an example:

```
"profileTopicMap": [  
  "customProfileClass": "MyCustomProfile"  
],
```

The `ercustomclass` attribute "customProfileClass" is associated with an object profile whose name can't be used as is when constructing a topic. And so you map "customProfileClass" to "MyCustomProfile" (the topic name you specify is case-sensitive). This means that any objects which ISIQ finds for "customProfileClass" will be placed in the "MyCustomProfile" topic for subscribers to consume. Also, because you've defined a mapping relationship between "customProfileClass" and "MyCustomProfile", you must add this mapping to the `txdef.json` file, as described earlier in this Appendix.

Restart the ISIQ Stacks

After you make changes in `isimConfiguration.json` or `txdef.json`, you must restart the ISIQ stacks to implement the changes. A restart entails removing all stacks and then deploying them again. The commands for restarting can be found in the [ISIQ Deployment Guide](#) sections, "Removing the ISIQ stacks" and "Deploying the ISIQ stacks".

Update the IGI-to-ISIM Subscription

After you make changes in `txdef.json` and restart the ISIQ stacks, you also need to update the IGI-to-ISIM subscription so that custom service-related entities are loaded into IGI. Follow these steps:

1. From the ISIQ UI, select your IGI product from the Control Center menu. You must navigate to your IGI product dashboard since you want to make the custom service-related topics from ISIM available to your IGI subscriber.
2. Go to the "Subscriptions" half of the product dashboard, find the IBM Security Identity Manager product that IGI is subscribed to, and then perform one or both of these update actions:
 - (a) If there's a new ISIM profile with associated topics that you want to load into IGI, click the "Update subscription" button (). That forces a re-read of all ISIM topics.
 - (b) If you changed the transformation rules for Person or BPPerson entities, click the "Reprocess" button  and then select the "*.Person" or "*.BPPerson" topics for reprocessing. That causes ISIM Person-related topic data to get replayed using your changed transformation rules when inserting the entities into IGI.

Appendix F: Removing ISIM Data in IGI

If you have an ISIQ environment that was used for ISIM-to-IGI data integration, and the environment gets reset (refer to the “[Docker Cleanup](#)” section of the [ISIQ Deployment Guide](#)), you might want to first remove previously integrated ISIM data from your IGI database in order to start over. A similar scenario occurs if you’re testing a custom transformation change in txdef.json and you want to clear out the IGI DB before performing a full ISIM-to-IGI data load to validate the txdef.json change. Note: These scenarios typically only arise in test and QA environments, and not in production.

Recommendations/Caveats:

- (1) The instructions in this Appendix assume you have a small-to-moderate number of entities to delete from IGI. If instead you have tens of thousands of users associated with an application and you attempt to delete the application in the IGI UI, the UI can hang while the background processing occurs. When you perform high-volume deletes, it is recommended that you not attempt to do so in the UI. One alternative is to use IGI’s Bulk Data Load tool to remove large numbers of users, applications, etc. As an example, see the IBM article, [Remove Users Record Track](#). The Bulk Load tool can be used in conjunction with the IGI Export Report feature to run queries that produce files, which become input to the Bulk Load tool. For more information, see “Export reports” in IGI’s [Available reports](#).
- (2) If you’re migrating from ISIGADI to ISIQ, you should **not** use these instructions to remove ISIGADI-synced entities from IGI before switching to ISIQ, and especially not if it’s a production IGI. There’s a risk that some of your ISIGADI-based customizations or other manual changes in IGI will be lost in the process. When migrating, it’s better to use the ISIGADI-to-ISIQ migration tool described in Appendix H.
- (3) If you don’t want the deletes in IGI to generate OUT events that flow back to ISIM, refer to the instructions in the IGI troubleshooting technote, [How to execute operations on Identity Governance & Intelligence \(IGI\), without triggering new events](#).

If you can revert to a snapshot of an empty, initialized IGI data tier, then that’s the quickest way to remove previously integrated ISIM data. But if you don’t have a usable snapshot, here is the sequence to follow when manually deleting IGI data that was ported from ISIM:

1. Applications
2. Account Configurations
3. Rights Lookups
4. Users
5. Organizational Units
6. Events for Account, Access, User, and Organizational Unit

The detailed steps are shown in the following sections.

Log in as Admin User

Log in to the IGI Administration Console as an administrator user. Note: By default, when a user is created in IGI, it’s a Service Center user. The ISIM data removal steps require an administrator user. For help with assigning an administrator role to an IGI user, refer to the appropriate IBM Documentation links, for example: [User administration](#).

Remove Applications

- Go to **Access Governance Core**.
- Go to **Manage -> Applications**.
- Select all the applications that were ported from ISIM.

NOTE: Do not select the **"ISIM"** application itself. From IGI's perspective, **"ISIM"** is an application that wasn't migrated from ISIM and therefore must not be removed.

Select **"ISIM – isimProductName"**. This application is created for ISIM organizational roles.

- Select **Remove** from the **Actions** menu.
When an application is removed, all permissions and external roles in that application are also removed.

Remove Account Configurations

When a service is ported from ISIM, an application and an account configuration are created, one per service. So, whenever you remove the application from IGI, you must also remove its account configuration.

- Go to **Access Governance Core**.
- Go to **Manage -> Account Configurations**.
- Select all account configurations that were created for ISIM services.
 - o NOTE: Do not select **Ideas** and **ISIM**. They are reserved applications in IGI and are not considered to be services ported from ISIM.
 - o Select **"ISIM – isimProductName"**. This application is created for ISIM organizational roles.
- Select **Remove** from the **Actions** drop-down menu.
 - o When an account configuration is removed, all IGI accounts in that account configuration are removed as well. The accounts in the target are not affected.

Remove Rights Lookups

When a permission with rights is ported from ISIM, a canonical list of lookup values is created. Thus, whenever you remove an application with this type of permission from IGI, you must also remove its Rights Lookup configuration:

- Go to **Access Governance Core**.
- Go to **Configure -> Lookup**.
- One by one, select each Right and then select **Remove** from the **Actions** drop-down menu.

NOTE: There are three reserved entries that must not be removed: 1_VPN, 2_EMAIL, 3_PRODUCTLINE.

Remove Users

- Go to **Access Governance Core**.
- Go to **Manage -> Users**.
- Select all users that were ported from ISIM.
- Select **Remove** from the **Actions** drop-down menu.

NOTE: The sequence of operations matters here. If Organizational Units are removed before Users, then all users will disappear from the IGI UI, but they will remain in the IGI database.

Remove Organizational Units

- Go to **Access Governance Core**.
- Go to **Manage -> Groups**.
- Select the parent OU that was ported from ISIM.

- Select **Remove** from the **Actions** drop-down menu.
- Select **Remove Node and Children** from the confirmation window.
- Select **OK**.

NOTE: After IGI organizational units are deleted from the IGI administration console, you need to also delete the OUs from the ORGANIZATIONAL_UNIT_ERC table. Delete only the ones ported over from ISIM. You can identify them by the <subscriberID>.<configurationName>: prefix in the OU column. For example, if the ISIQ subscriber ID was “scharle” and the configured IBM Security Identity Manager product name was “Alpha”, then an SQL select command such as **select id, name, ou from igacore.organizational_unit_erc where ou like 'scharle.Alpha%';** shows the following:

ID	NAME	OU
139	IBM Costa Mesa	scharle.Alpha:erglobalid=7041375949262930062,ou=org,dc=com
140	sub org	scharle.Alpha:erglobalid=4206897811063818855,ou=orgChart,erg
141	Organization	scharle.Alpha:erglobalid=00000000000000000000,ou=org,dc=com
142	IIES Dev	scharle.Alpha:erglobalid=7859961573191755346,ou=orgChart,erg
143	Black Hats	scharle.Alpha:erglobalid=7862994008640012864,ou=orgChart,erg
180	Consultants	scharle.Alpha:erglobalid=7856543942757875291,ou=orgChart,erg

Then, run a Delete SQL command to remove just those rows.

Remove Events

- Go to **Access Governance Core**.
- Go to **Monitor -> Target inbound – Account events**.
- Select all account events for the application ported from ISIM and remove them.
 - o Go to **Actions -> Remove**.
 - o If you want to remove all the ones shown on this page, go to **Actions -> Select Delete all filtered**.
- Also remove events for that ISIM in the following tabs:
 - o Target inbound – Access events
 - o IN – User events
 - o IN – Org. Unit events

Once you’ve completed the deletes and have an empty IGI database, it’s a good practice to take a snapshot or backup of the IGI data tier. This will allow you to quickly restore the initialized environment in case you need to rerun the ISIM-to-IGI data load.

Reloading ISIM Data

After you restore an initialized IGI database and you’re ready to reload ISIM data, you have two choices:

1. Run ``docker volume prune --all``. This is a powerful command that should be used with caution. It erases all persistent ISIQ data, including product configurations, subscriptions, user groups, audit events, and Kafka topic data. You then have to reconfigure your ISIM and IGI products, and re-subscribe IGI to ISIM. Doing so causes another data load to occur. For a fuller discussion of ``docker volume prune --all``, see the “Docker Cleanup” section in the [ISIQ Deployment Guide](#).
2. A more granular approach is to run the `<starterKitDir>/util/topicList.sh` script to list all topics. Next, run `<starterKitDir>/util/topicDelete.sh <topic_name>` to remove each topic belonging to

the IGI product you want to reload. These topics have “IGI_database” in their name (note: you might need to stop ISIQ’s app and connect stacks before the `topicDelete.sh` command takes effect, if the connect stack still has a hold on the IGI topics). Then, go to the IGI product dashboard page and click the “Reprocess” button for your IGI-to-ISIM subscription. Select all ISIM directory topics for reprocessing. By performing these steps, you will replay the directory source topics from the beginning, which in effect reloads your ISIM data into IGI.

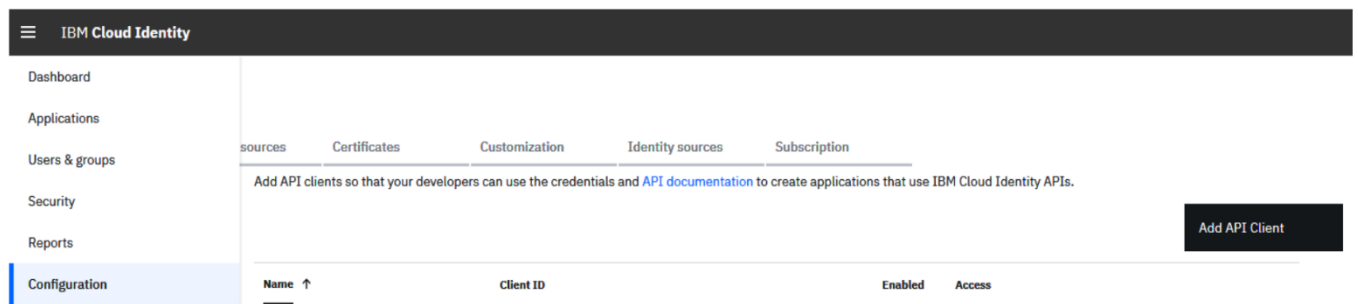
Appendix G: ISIM Integration with Cloud Identity

ISIQ supports an experimental ISIM-to-IBM Cloud Identity (CI) integration in which ISIM Persons are imported into CI as Users. There are default transformation rules defined in txdef.json that map ISIM Person and BPPerson entities to CI Users.

Since this ISIQ feature was introduced, IBM Cloud Identity has been rebranded as IBM Security Verify, and so to learn more about IBM's Identity-as-a-Service (IDaaS) offering, search the IBM Docs for "Security Verify".

Here are the steps to follow to try out this integration:

1) Navigate to the "Configuration" page in your CI instance.



2) Click the "Add API Client" button and register ISIQ as an API Client. Grant ISIQ access to All APIs by toggling "Select All". Select "Save" when finished.

Add API Client

Name*

Enabled ☒ On

Credentials

Client ID

Client Secret

Access

Select the APIs that you want to grant access:

Select All ☒ On

- ☒ Authenticate any user
- ☒ Generate OTP
- ☒ Manage access policies
- ☒ Manage API clients
- ☒ Manage application entitlements
- ☒ Manage application lifecycle
- ☒ Manage attribute sources
- ☒ Manage authenticator configuration
- ☒ Manage authenticator registrations for all users

3) Edit the created API Client instance (using the Pencil icon) and take note of the “Client ID” and “Client Secret” fields.

Edit API Client

Name*

Enabled ☒ On

Credentials

Client ID

Client Secret Reveal

Access

Select the APIs that you want to grant access:

Select All ☒ On

- ☒ Authenticate any user
- ☒ Generate OTP
- ☒ Manage access policies
- ☒ Manage API clients
- ☒ Manage application entitlements
- ☒ Manage application lifecycle
- ☒ Manage attribute sources
- ☒ Manage authenticator configuration
- ☒ Manage authenticator registrations for all users

Cancel Save

4) Configure your application as an ISIQ product by choosing “IBM Cloud Identity (experimental)” from the “Select a Product” page:

Select a product to configure from the list of available products

IBM Security Identity Manager	https://www.ibm.com/support/knowledgecenter/en/SSRMWJ_7.0.1/com.ibm.isim.doc
IBM Security Verify Governance	https://www.ibm.com/us-en/marketplace/identity-governance-and-intelligence
External Application	https://www.ibm.com/support/pages/ibm-security-information-queue-users-guide
IBM Cloud Identity (experimental)	https://www.ibm.com/us-en/marketplace/cloud-identity

Cancel < Back Next >

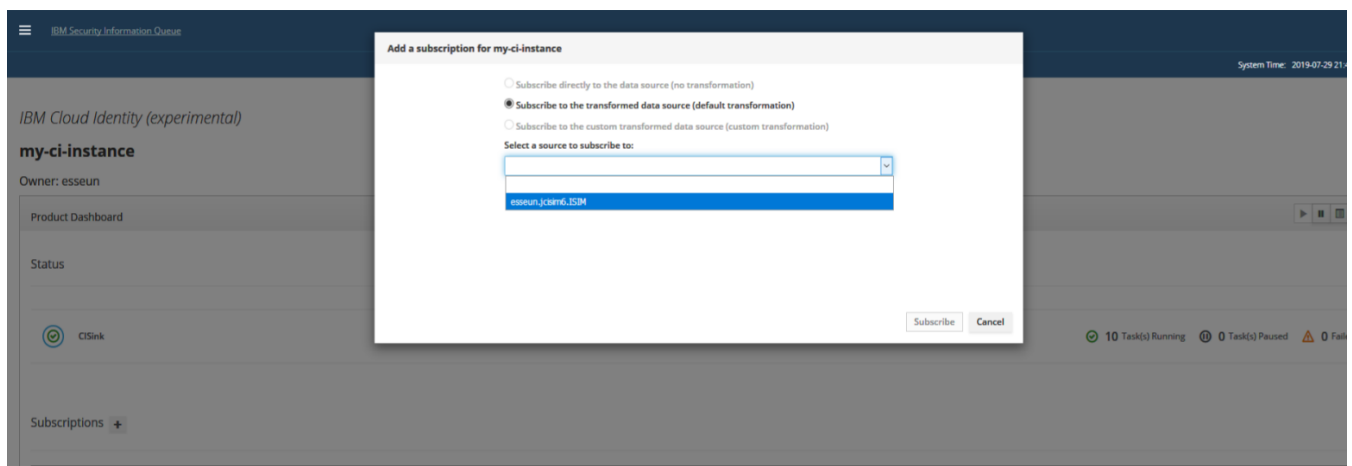
5) Complete the required fields, with values noted from Step 3)

Configure IBM Cloud Identity (experimental)

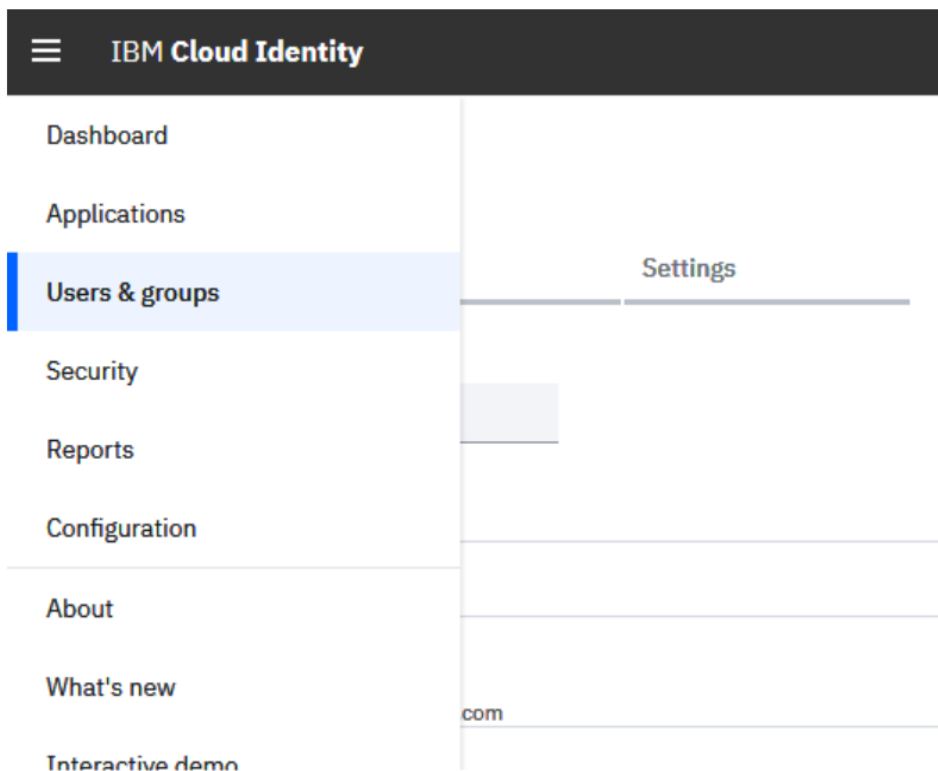
Configuration Name my-config Tenant URL https://user.ice.ibmcloud.com:443	Description (optional) description of product being configured Client ID a12b3456-c78d-9101-e112-1f3g1h41ij5k Client Secret
---	--

After adding your Client Secret and clicking “Configure”, your CI product should be registered with ISIQ.

6) Navigate to the ISIQ dashboard of your CI product (called “my-ci-instance” in this example) and create a subscription to your configured IBM Security Identity Manager product.



7) Navigate to “Users & groups” in your CI instance to verify that ISIM Persons are being inserted.



Note: ISIM-to-CI is currently a one-way integration that imports ISIM Persons as CI Users. Other entity integrations and operations between CI and ISIM are not yet supported.

Appendix H: Optional ISIQ Tools

This Appendix describes optional tools that ISIQ offers to let you perform assorted utility functions. The tools are available in the isiq-sdi-tool zip file that you can download from the [ISIQ Starter Kit Page](#).

ISIM-to-IGI Validation Tool

After you subscribe IGI to ISIM, you will likely want to validate whether all ISIM entities (services, groups, users, accounts, and roles) were loaded successfully into IGI. To satisfy this requirement, ISIQ provides a containerized IBM Security Directory Integrator (SDI) server that runs assembly lines to check if each ISIM entity exists in IGI with the correct key. The results are reported in docker log messages so that you can validate if data integration between the products is complete.

The steps for configuring and running the validation tool are explained in
<sdiToolDir>/isiq-sdi-tool/README.txt

If the tool finds that any ISIM entities were not integrated into IGI, refer to the ISIQ diagnostic procedures described in the [ISIQ Troubleshooting Guide](#).

ISIGADI-to-ISIQ Migration Tool

To help ISIGADI users migrate to ISIQ, there is a migration tool that runs, like the validation tool, in the containerized SDI. The procedure for configuring and running the migration tool is described in
<sdiToolDir>/isiq-sdi-tool/README.migrate.txt

One of the last steps in the procedure is to “Set ISIM LDAP source offset and IGI DB source offset.” The reason for performing this step is the following: Suppose you have 200K ISIM accounts that ISIGADI loaded into IGI. You stop ISIGADI, run the migration tool to convert the 200K accounts to ISIQ format, and then you subscribe IGI to ISIM from the ISIQ UI. By default, ISIQ will process all of those accounts again because from the standpoint of the Kafka-based Account source topic, ISIQ is starting at offset 0, which means it reads LDAP from the beginning. Consequently, a lot of unnecessary overhead occurs as ISIQ reprocesses account data already stored in IGI.

To prevent this scenario from happening, you should determine the timestamp of the last ISIGADI entry which was sync’d into IGI, and then specify that timestamp as an offset in ISIQ. Configuring a timestamp offset causes ISIQ to begin processing ISIM add/change/delete events **after** the offset you specify.

The same mechanism works for integrating IGI events to ISIM after running the migration tool. In that circumstance, you don’t want the same IGI events to be integrated to ISIM again, and so you set offsets equal to the last Event ID numbers that were processed.

Here are the instructions for setting offsets:

Setting Initial Offsets for ISIM and IGI Source Topics

As of ISIQ 1.0.7, you can set initial offsets for ISIM and IGI source topics. The offsets are used in the ISIQ connector logic to indicate which entities have previously been processed for a topic. The format of the offset will vary depending on the entity type.

For ISIM-to-IGI integration, the LDAP timestamp of ISIM entities is used as the offset. The LDAP timestamp must be in the format of yyyyymmddhhmmssZ, where “yyyyymmdd” is the date using a 4-digit year, 2-digit month, and 2-digit day; and “hhmmss” is the time using a 2-digit hour, 2-digit minute, and 2-digit seconds. The timestamp must be GMT or UTC time, also known as Zero Meridian or Coordinated Universal Time. For example, if you are located in the Pacific time zone, which is UTC-8, and you want to

set the offset to the current time, add 8 hours to the current time. You also need to set the date according to the current UTC date. When a timestamp offset is specified, only ISIM entities created or updated **AFTER** that timestamp will be populated in LDAP source topics to be consumed by IGI.

For IGI-to-ISIM integration, the IGI event's unique ID serves as the offset. The unique ID is the integer designated as the primary key in IGI's event tables. When integrating IGI events to ISIM, ISIQ reads account events and user events. Account events come from the USER_EVENT_ERC table, and user events come from the EVENT_OUT_USER table. When you specify the offset in ISIQ, only IGI events that have an ID **GREATER** than the offset ID will be populated in the IGI source topics. For example, if you set the offset ID as 2999, then all events with ID 3000 or greater will be populated in IGI source topics to be consumed by ISIM.

To set the initial offsets for ISIM and IGI source topics, you must set the following environment variables in your connect-stack.yml file:

- IQ_ISIM_LDAP_OFFSETS_AFTER_ISIGADI_MIGRATION
- IQ_IGISRC_USER_EVENT_ERC_OFFSET
- IQ_IGISRC_EVENT_OUT_USER_OFFSET

If you have a cluster environment, set these environment variables, with the same values, in all of your connect-stack.yml files.

These environment variable offsets are read if and only if non-zero offsets do not exist in the Kafka offset topics. In other words, environment variable offsets are used only once after ISIQ's connect stack is started. After new entities begin to be processed from a source topic, the offset of the last entity processed becomes the new offset in the Kafka offset topic. When that occurs, the offset environment variables are no longer required and you can safely remove them from connect-stack.yml.

Here is a more detailed explanation of the three environment variables:

IQ_ISIM_LDAP_OFFSETS_AFTER_ISIGADI_MIGRATION

This environment variable sets the timestamp offset for ISIM LDAP source topics. The timestamp must be in the format of <isimProductName>;;<ldapTimeStamp>. If you want to set offsets for multiple IBM Security Identity Manager products, list the multiple timestamps with a comma (,) separator. The "ldapTimeStamp" must have the format, yyyyymmddhhmmssZ, where the timestamp is specified in Coordinated Universal Time.

Example 1: If you configured an IBM Security Identity Manager product as "isim6" in ISIQ, and you want to set its LDAP offset to 05/10/2020 12:00:00 in Coordinated Universal Time, set the environment variable as follows:

- IQ_ISIM_LDAP_OFFSETS_AFTER_ISIGADI_MIGRATION=isim6;;;20200510120000Z

LDAP entities created or updated after 05/10/2020 12:00:00 will be populated in ISIM source topics.

Example 2: If you configured two IBM Security Identity Manager products, (1) "isim6" whose LDAP offset you want to set to 04/28/2020 11:00:00 Coordinated Universal Time; and (2) "isim7" whose LDAP offset you want to set to 05/10/2020 05:30:00 Coordinated Universal Time, then set the environment variable as follows:

- IQ_ISIM_LDAP_OFFSETS_AFTER_ISIGADI_MIGRATION=isim6;;;20200428110000Z,isim7;;;20200510053000Z

IQ_IGISRC_USER_EVENT_ERC_OFFSET

This environment variable sets the offset for IGI's USER_EVENT_ERC table, where account-related events are stored.

Example 1: If you configured an IGI product called "isvg1001", and you want to process only the events with an ID greater than 4090, set the environment variable as follows:

- IQ_IGISRC_USER_EVENT_ERC_OFFSET=isvg1001;;;4090

Account events in the IGI OUT event queue with an ID of 4091 or greater will be populated in the IGI source topic.

Example 2: If you configured two IGI products, "isvg1001" and "isvg1001fp1", with offsets 4090 and 2100, then set the environment variable as follows:

- IQ_IGISRC_USER_EVENT_ERC_OFFSET=isvg1001;;;4090,isvg1001fp1;;;2100

IQ_IGISRC_EVENT_OUT_USER_OFFSET

This environment variable set the offset for IGI's EVENT_OUT_USER table, where user-related events are stored.

Example 1: If you configured an IGI product called "isvg1001", and you want to process only the events with an ID greater than 1999, set the environment variable as follows:

- IQ_IGISRC_EVENT_OUT_USER_OFFSET=isvg1001;;;1999

User events in the IGI OUT event queue with an ID of 2000 or greater will be populated in the IGI source topic.

Example 2: If you configured two IGI products, "isvg1001" and "isvg1001fp1", with offsets 1099 and 2999, then set the environment variable as follows:

- IQ_IGISRC_EVENT_OUT_USER_OFFSET=isvg1001;;;1099,isvg1001fp1;;;2999