

Migrating a connector from previous versions using the Data Integration REST API

Abstract

This article describes how to use the Informatica REST API to migrate an old version of the Cloud Data Integration connector to the current version and retain the referenced assets after the migration.

Supported Versions

- Informatica Cloud® Data Integration Connectors

Table of Contents

Overview	2
Compatibility with earlier versions.	3
Planning the connection migration	3
Migration process.	4
View the updated assets.	5
Formats to use with the REST API version 2.	5
login.	6
connectionMigration.	8
Starting a migrate job.	8
logout.	10
Migration guidelines and restrictions.	10
Amazon Redshift.	12
Amazon S3.	13
Microsoft Azure Blob Storage.	15
Microsoft Azure Synapse SQL.	17
Snowflake.	20
Troubleshooting the migration.	20

Overview

You can use the Informatica Intelligent Cloud Services REST API to seamlessly migrate from an earlier version of the connector to the current available version. The assets such as mappings and mapping tasks that reference the old connection version updates to the current version.

To perform the migration, you need access to the connector and the referenced assets in your Informatica Intelligent Cloud Services organization. To configure a request using the REST API, use the appropriate resource and method, along with the applicable objects. Informatica Intelligent Cloud Services returns the requested information, performs the requested task, or returns an error and related messages.

You can migrate the following version of the connectors to the current version:

Old version	Current version
Amazon Redshift V1 Connector	Amazon Redshift V2 Connector
Amazon S3 V1 Connector	Amazon S3 V2 Connector
Microsoft Azure Blob Storage V1 Connector	Microsoft Azure Blob Storage V3 Connector
Microsoft Azure Blob Storage V2 Connector	Microsoft Azure Blob Storage V3 Connector
Microsoft Azure SQL Data Warehouse V1 Connector	Microsoft Azure Synapse SQL Connector
Microsoft Azure SQL Data Warehouse V2 Connector	Microsoft Azure Synapse SQL Connector
Snowflake Cloud Data Warehouse Connector	Snowflake Data Cloud Connector

Compatibility with earlier versions

The old version of the connection that you want to migrate might be used in one or more mappings or mapping tasks. The REST API identifies all the assets configured with the connection that you want to migrate and updates those assets without disrupting the assets of other connections.

Only Source, Target, and Lookup transformations that reference the old connection are applicable for the connection migration. Other transformations such as SQL transformation that reference the old connection are not applicable. You also cannot migrate connections used in assets such as synchronization tasks or mappings in advanced mode.

Before migrating the connection to the current version, you must consider the following factors that might affect the migration:

- The connection type from the older to the current version must be compatible. If the source and target connectors are different, the configured advanced connection properties from the old connector might be lost during the migration process.
- The configured connection attributes in the older version of the connection must be compatible with the current version.

Planning the connection migration

You must plan your migration before you can start the migration programmatically.

Prerequisites

Complete the following prerequisites before you start the migration:

- You must have the licenses to import, export, create, and delete projects.
- You need the license for the latest version of the connector that you plan to migrate to.
- To use the Informatica Intelligent Cloud Services REST API, you need a valid Informatica Intelligent Cloud Services login and an understanding of the REST API guidelines.
- Ensure that metering is enabled for the **Total Number of Projects** and **Total Number of Folders** for the platform service type under **License Metrics** on the **Metering** tab in the Administrator service.
- You need to install a REST client tool such as Postman.

Set up the migration environment

Set up the migration environment in Cloud Data Integration:

- Keep the connector and dependent assets ready for migration in your organization.
- Create the current version of the connection for the migration and ensure that the test connection runs successfully. During the migration, this connection replaces the older connection in the referenced assets.
- Ensure that the current version of the connection along with the referenced assets and the old version of the connection are in the same organization.

The next thing we'll need to do is log in to the organization using the Informatica Intelligent Cloud Services REST API and begin the migration.

Migration process

The migrate API provides a simple way to migrate your connections in the configured assets to the current version.

After you complete the prerequisites, perform the following tasks to migrate the connection:

1. Using a REST API client tool, log in to your organization that contains the older version of the connector that you want to migrate.
 - a. Send a login POST request using the following URL for your login resource: `https://dm-<POD region>.informaticacloud.com/ma/api/v2/user/login`
 - b. Specify the Informatica Intelligent Cloud Services user account and password attributes in the login object.

The call returns the user object if the request is successful. Returns the error object if errors occur.

The response includes the following information that you need to include in the header of the migrate REST API call:

 - `icSessionId`. A REST API session ID that you include in the header for version 2 REST API calls.
 - `serverUrl`. The base URL that you can use in version 2 resource URIs, for example: `<serverUrl>/saas/api/v2/job`

Use the base URL and session ID returned in the response for the subsequent migrate requests during this session. For more information, see ["login" on page 6](#)
2. You can use the objects resource to get a list of asset dependencies of the connection before you start the migration. For more information, see [Finding asset dependencies](#).
3. Send a migrate POST request to migrate your assets referenced with the old connection to use the new connection.
 - a. Use the following URL for your migrate resource: `https://<serverURL>/saas/api/v2/connectionMigration/migrate`

For example, the `serverUrl` is `https://na4.dm-us.informaticacloud.com/saas` and the URI is `/api/v2/connectionMigration/migrate`

- b. In the request header, specify the `icSessionId`.
 - a. In the request body, specify following attributes:

`sourceConn`. Name of the older version of the connection that you want to migrate.

`targetConn`. Name of the current version of the connection to which you want to migrate.

`projectName`. Name of the project folder that contains the referenced assets of the source connection. If you do not specify the folder name, the migrate resource considers assets that reference the source connection from all folders.

The call returns the migrate job ID, migrate job name, and the source and target folder paths in Cloud Data Integration. The source path contains the old connector version with the referenced assets, while the target path contains the assets updated to use the new current connector version. The target folder path uses the same name as the source and has the migration job ID appended to the name.

For more information on how to use the migrate API call, see [“Starting a migrate job” on page 8](#).

4. Under projects in **Explore** in Cloud Data Integration, search for the target folder name and verify if the assets are updated to use the current version of the connector.

View the updated assets

If the assets that reference the old connection are in a single or multiple folders in Cloud Data Integration, the migrate API creates a corresponding new folder or folders for the updated assets when the migration is complete.

The migration call returns the migrate job ID, migrate job name, and the source and target paths in Cloud Data Integration. The source path contains the assets referenced with the old connection, while the target path contains the assets updated with the current connection version. The target folder name uses the same name as the source, but has the migration job ID appended to the name.

For example, if the source folder with the referenced assets is `AA_OldArtifacts`, the updated assets are copied to a target folder `AA_OldArtifacts_Migration_1ABC0`, with the migration job ID is suffixed to the target folder.

If the assets that reference the source connection are in multiple folders, the migrate API creates corresponding new folders to include the updated assets.

For example, if the folder names that contain the assets is `AA_OldArtifacts1` and `AA_OldArtifacts2`, these assets are copied to corresponding new folders `AA_OldArtifacts_Migration_1ABC0` and `AA_OldArtifacts_Migration_2ABC0`.

If the assets in the source path are at the project level and not within a folder in a project, the target path folder name takes the source project name.

When you click the migrated asset in the folders, the names of these assets, for example, the mapping name, remains the same as in the old connection, but the old connection in the assets is replaced by the current version.

Formats to use with the REST API version 2

When you use the Informatica Intelligent Cloud Services platform REST API resources version 2, use the appropriate format for the resources.

You can log in to Informatica Intelligent Cloud Services using the platform REST API version 2 login resource.

Format

You can use the following XML and JSON call formats for REST API version 2.

Login URL

Use one of the following login URLs: `https://dm-<POD region>.informaticacloud.com/ma/api/v2/user/login`.

Your POD (Point of Deployment) region is based on the location of your Informatica Intelligent Cloud Services data center. Use one of the following POD regions:

- For North America, use `us`
- For Europe, use `em`
- For Asia, use `ap`

The POD region is included in the URL you receive when you register with Informatica Intelligent Cloud Services.

Base URL

The login response includes the base URL that you must include in subsequent calls.

The base URL includes the following components:

- The name and region of the POD that your organization uses, for example, `usw3.dm-us`.
- The Informatica Intelligent Cloud Services domain, `informaticacloud.com`.
- The internal service that manages the API calls.

The following example is a base URL for an organization on the `usw3.dm-us` POD:

```
https://usw3.dm-us.informaticacloud.com/saas
```

In the login response, the attribute name is `serverUrl`.

Request URL

The URL that you can use in requests is: `<serverURL>/saas/api/v2/<REST API Resource>`

For example:

```
https://usw3.dm-us.informaticacloud.com/saas/api/v2/connectionMigration
```

Session ID

The login response includes a session ID that you must include in headers during the session. You can use `icSessionId` in the header.

login

You can use this resource to log in to your organization using your Informatica Intelligent Cloud Services user account.

Use the base URL and session ID returned in the response for subsequent requests during this session.

POST request

To log in, use the following URL:

```
http://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/login
```

The values for cloud provider and region correspond to the name of the POD (Point of Deployment) that your organization uses.

For example, if your organization uses the APNE1 POD, use the following URL:

```
http://dm1-ap.informaticacloud.com/ma/api/v2/user/login
```

If you don't know the name of the POD that your organization uses, contact your organization administrator or Informatica Global Customer Support.

For more information about the POD names and corresponding cloud providers and regions, see the Product Availability Matrix (PAM) for Informatica Intelligent Cloud Services.

Use the following attributes in a login object:

Field	Type	Required	Description
username	String	Yes	Informatica Intelligent Cloud Services user name. Maximum length is 255 characters.
password	String	Yes	Informatica Intelligent Cloud Services password. Maximum length is 255 characters.

POST request example

To log in to your Informatica Intelligent Cloud Services organization, you might use the following request:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/login
Content-Type: application/json
Accept: application/json

{
  "username": "John@infa.com",
  "password": "mypassword"
}
```

POST response

Returns the user object if the request is successful. Returns the error object if errors occur.

The response includes the following information that you need to include in the header of subsequent REST API calls:

- **icSessionId.** A REST API session ID that you include in the header for version 2 REST API calls. The session ID expires after 30 minutes of inactivity. After the session ID expires, log in again to continue working with the REST API.
For information on retrieving session status details, Session IDs.
- **serverUrl.** The base URL that you use in the version 2 resource URLs.

For information about the attributes that the user object includes, see [login](#).

POST response example

The response returns the user object which contains the serverUrl and icSessionId values to use in subsequent calls, as shown in the following example:

```
{
  "id": "0101TQ03000000000007",
  "orgId": "0101TQ",
  "orgUuid": "3FNFLsluHe2IIgTs8tRjSJ",
  "name": "John@infa.com",
  "description": "",
  "createTime": "2018-02-16T00:20:07.000Z",
  "updateTime": "2018-07-17T22:45:50.000Z",
  "createdBy": "System built-in user",
  "updatedBy": "John@infa.com",
  "sfUsername": null,
  "firstName": "John",
  "lastName": "Randall",
  "title": "IICS Admin",
  "password": "*****",
  "phone": "123-456-7899",
  "emails": "John@infa.com",
  "timezone": null,
  "serverUrl": "https://na4.dm-us.informaticacloud.com/saas",
}
```

```

    "icSessionId": "1Ckv5VDHe2IICHi2hq04EF",
    "securityQuestion": "In what city were you born?",
    "securityAnswer": "*****",
    "uuid": "a51jk7TB0IDcnWLwJdLaW2",
    "forceChangePassword": false,
    "roles": [
      {
        "name": "Designer",
        "description": "Role for creating assets, tasks, and processes. Can configure
connections, schedules, and runtime environments. Has access to the Application Integration
Console."
      },
      {
        "name": "Admin",
        "description": "Role for performing administrative tasks for an organization. Has
full access to all licensed services."
      },
      {
        "name": "Data Preview",
        "description": "Role to preview data"
      }
    ],
    "usergroups": [],
    "spiUrl": "https://na4.dm-us.informaticacloud.com"
  }
}

```

Using the above response as an example, to send a POST request to migrate the connection from the old to the current version, you might use the following request:

```

GET https://usw3.dm-us.informaticacloud.com/saas/api/v2/agent
Content-Type: application/json
Accept: application/json
icSessionId: 1Ckv5VDHe2IICHi2hq04EF

```

connectionMigration

Use this resource to update assets such as mappings and mapping tasks that reference the old version of the connector to the current version of the connector.

If the connection is used across multiple assets, all the referenced mappings and tasks are updated to use the current version. The migration creates a copy of the project folder and assets and updates the referenced connection in the assets with the latest version.

Starting a migrate job

Use a POST request to start a migrate job.

POST request

To specify the connections to migrate and start the migrate job, use the following URI:

```
saas/api/v2/connectionMigration/migrate
```

Field	Type	Required	Description
icSessionId	String	Yes	Informatica Intelligent Cloud Services session ID.
sourceConn	String	Yes	The name of the source connection that you want to migrate. This is the old version of the connector.

Field	Type	Required	Description
targetConn	String	Yes	The target connection name to which you want to migrate. This is the current version of the connector.
projectName	String	No	The project name contains the referenced assets of the source connection that you want to migrate. If you do not specify the project name, the source connection in all assets are migrated.

Include the following fields in the request:

POST request example

You can use a request similar as the following example:

```
POST <baseUrl>/saas/api/v2/connectionMigration/migrate
Content-Type: application/json
Accept: application/json
icSessionId: <sessionId>

{
  "sourceConn": "Migration_V1",
  "targetConn": "Migration_V2",
  "projectName": "V1_Assets"
}
```

POST response

If successful, returns the following information for the export job:

Field	Type	Description
id	String	Global unique identifier for the migration job.
name	String	Name of the migration job.
count	Integer	Number of dependent objects.
status	Complex type	Status of the migration.
objects	Collection	Objects in the migration job.
name	String	Returned in the objects object. Name of the asset that uses the connector.
type	String	Returned in the objects object. The type of asset.
sourcePath	String	Returned in the objects object. Complete path of the referenced asset in the old connection.
targetPath	String	Returned in the objects object. Complete path of the referenced assets in the current connection. The target folder uses the same name as the source folder, with "_Migration_<migration job ID>" appended to the name for identification.
status	String	Returned in the objects object. State of the connection updated in the asset, such as IN PROGRESS, SUCCESS, or FAILED.

POST response examples

If successful, you might receive a response similar to the following example:

```
{
  "id": "411XcURpYFjjxpopb7AJ0L",
  "name": "job-1661702962208",
  "count": "1",
  "status": "SUCCESSFUL",
  "objects": [
    {
      "name": "Mapping_V1",
      "type": "DTEMPLATE",
      "sourcePath": "/AA_Migration_Doc",
      "targetPath": "/AA_Migration_Doc_Migration_411XcURpYFjjxpopb7AJ0L",
      "status": "SUCCESSFUL"
    }
  ]
}
```

If you receive an error, you might see a response similar to the following example:

```
{
  "error": {
    "code": "MigrationSvc_034",
    "message": "Invalid object id/s [[242973wgfscbwad23]]. Object resolution failed.",
    "requestId": "2ataxVlgw3ydI1Yb2MA4sq"
  }
}
```

logout

Use this resource to log out of an organization and end the version 2 REST API session specified in the request.

POST Request

To log out an organization and end the version 2 REST API session, include the Informatica Intelligent Cloud Services session ID in the request header with the following URI.

```
saas/api/v2/user/logout
```

POST Response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST Example

To log out of your Informatica Intelligent Cloud Services organization, you might use the following request:

```
POST <baseUrl>/saas/api/v2/user/logout
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>
```

Migration guidelines and restrictions

This section includes the connections that you can migrate and the properties in the referenced assets of the connection that are applicable for migration. This section also describes guidelines and restrictions that are applicable for the connection migration.

When you migrate, the configured advanced properties and the field mappings in mappings and tasks from the old connection version are retained in the current version. However, due to differences between the old and current

connection versions, some transformations, fields, functions, expressions, or data types might not be valid or might be valid with restrictions for the migration.

Consider the following general guidelines when you migrate a connection to the current version:

- If a connection is used in synchronization tasks or mappings configured in advanced mode, these assets are not updated with the new connection.
- You can migrate a connection referenced in multiple assets in one migration call. However, you cannot migrate connections when multiple transformations in the asset use different connection names or different connection types. If the connection names or types used in an asset are not the same, you need to migrate each transformation in the asset in different migration calls.
For example, you have an Amazon S3 V1 mapping that consists of a connection named S3V1 in the Source transformation and a separate connection named S3_V1 in the Target transformation. When you post a request using the migration API for the connection name, S3V1, the mapping is updated, where the Source transformation now references the current connection. To update the same asset that uses the S3_V1 connection in the Target transformation, you need to post another request to migrate the S3_V1 connection.
- If the old connection is referenced in a completely parameterized mapping, migration applies only if the asset uses the following parameterization combinations:
 - The mapping uses a connection and object, where both are parameterized.
 - The mapping uses a connection that is not parameterized, while the object is parameterized.

Note: If the mapping contains a parameterized connection, but if the object is not parameterized, the old connection in the asset is not switched with the new connection.

Amazon Redshift

The following table lists the configured advanced source, lookup, and target properties from Amazon Redshift V1 mappings that you can retain when you migrate to Amazon Redshift V2:

Transformations	Amazon Redshift V1 properties retained after migration to Amazon Redshift V2
Source and Lookup	<ul style="list-style-type: none"> - S3 Bucket Name - Enable Compression - Staging Directory Location - Unload Options - Encryption Type - Download S3 Files In Multiple Parts - Source Table Name - Pre-SQL - Post-SQL - SQL-Query - Select Distinct
Target	<ul style="list-style-type: none"> - S3 Bucket Name - Enable Compression - Staging Directory Location - S3 Client Side Encryption - S3 Server Side Encryption - Pre-SQL - Post-SQL - Batch Size - Number of Files per Batch - Max Errors Per Batch Upload Insert - Truncate Target Table Before Data Upload - Require Null Value For Char And Varchar - Wait Time In Seconds For S3 File Consistency - Copy Options - Vacuum Target Table - Analyze Target Table - Prefix to retain staging files on S3 - Target Table Name - Minimum Upload Part Size - Success File Directory - Error File Directory - Transfer Manager Thread Pool Size

Guidelines to migrate to Amazon Redshift V2

Some guidelines and restrictions apply to assets when you migrate the Amazon Redshift V1 connection to Amazon Redshift V2.

Source transformation

Consider the following migration guidelines when you configure a read operation:

- You cannot apply migration for mappings configured with the multi-object source type. When you migrate to Amazon Redshift V2, the mapping fails to read from multiple Amazon Redshift source objects and displays the following error: `Operation failed: error [String index out of range: -1]`
- When you run an Amazon Redshift V1 mapping with an SQL query to read from a date/time column, the mapping runs successfully. The same mapping fails after the migration to Amazon Redshift V2 with the following error:

```
[FATAL] Timestamp format must be yyyy-mm-dd hh:mm:ss[.fffffffff]
```

- If you specify a value for the **Number of Sorted Ports** field in the advanced properties in an Amazon Redshift V1 source object, this value does not apply for Redshift V2 after the migration. To configure a sort operation, you can use the **Sort** property from the **Query** tab in the Amazon Redshift V2 Source transformation after the migration.
- When you enable the **Turn on client side encryption** property to define the source attributes to read from an Amazon Redshift V1 source, these configured values are not retained after migrating to Amazon Redshift V2.
- When a mapping contains a parameterized Lookup transformation, the configured encryption in the connection used in the lookup is not retained after the migration.
- When you migrate and the mapping is configured with partitioning for the source, the number of rows processed by each partition varies in Amazon Redshift V2 when compared to Amazon Redshift V1.

Target transformation

Consider the following migration guidelines when you configure a write operation:

- When you configure any operation other than insert, and then migrate to Amazon Redshift V2, you must select the applicable target attribute **Treat Source Row As** in the Amazon Redshift V2 target properties before you run the mapping.
- If a mapping is configured with an override to the Amazon Redshift table where the overridden columns are of greater precision, the entire column with higher precision is skipped and an exception is logged. The difference in precision between the columns at design and runtime does not affect the same mapping that uses the Amazon Redshift V1 connection.
- The attribute **Number of bytes needed to support multibytes for varchar** is available in the Amazon Redshift V1 connection but not in Amazon Redshift V2. When you migrate to Amazon Redshift V2, the data appears truncated in the target columns. As a workaround, you can edit the metadata of the target before you run the mapping.

Parameterization

Consider the following migration guidelines when you parameterize mappings:

- If you create a completely parameterized mapping, do not specify the connection type in the connection parameterization. If you specify the connection type in the connection parameterization and then migrate the connection, the mapping fails at runtime.

Amazon S3

The following table lists the configured advanced source and target properties from Amazon S3 V1 mappings that you can retain when you migrate to Amazon S3 V2:

Transformations	Amazon S3 V1 properties retained after migration to Amazon S3 V2
Source	<ul style="list-style-type: none"> - Part Size - Staging Directory
Target	<ul style="list-style-type: none"> - Part Size - Staging Directory - TransferManager Thread Pool Size - Object Tags - Merge Partition Files - Folder Path - Compression Format - Encryption Type

Guidelines to migrate to Amazon S3 V2

Some guidelines and restrictions apply to assets when you migrate the Amazon S3 V1 connection to Amazon S3 V2.

Source transformation

Consider the following migration guidelines when you configure a read operation:

- When you create an Amazon S3 V1 mapping with a header file and set the advanced source properties **Header Line Number** and **Read Data from Line** as 0, the mapping fails after you migrate to Amazon S3 V2. This issue occurs because of the configuration differences for this functionality between Amazon S3 V1 and V2 connections. To read a file using Amazon S3 V1 Connector, you need to update the advanced source property **Read Data from Line** as 0 to read data from a file with a header. In Amazon S3 V2, you need to update the **Header Line Number** and **First Data Row** as 1 in the source formatting options.
- If you enable the **Read Data from Line** property advanced source property in an Amazon S3 V1 Source transformation, after the migration the header data is lost and replaced with `FIELD0`, `FIELD1`, `FIELD2`, `FIELD3`.
- The **Enable Downloading S3 Files in Multiple Parts** advanced source attribute that is available in Amazon S3 V1 is unavailable in Amazon S3 V2. If you enable the option to download the S3 file in multiple parts using the **Enable Downloading S3 File in Multiple Parts** field and also specify the part size to download the file in multiple parts in Amazon S3 V1, and you then migrate to Amazon S3 V2, the functionality to download in multiple parts is enabled through the **Multi-part Download Threshold** field. The configured part size is also retained in the **Download Part Size** field in the Amazon S3 V2 mapping.

Target transformation

Consider the following migration guidelines when you configure a write operation:

- The following restrictions when you run a mapping configured to create a new target at runtime:
 - Special characters from the source appear without the double quotes in the target.
 - The target output file is not uploaded to the specified Amazon S3 bucket path. The file is uploaded to a path that is not valid.
 - Junk characters appear in the object name after the migration. The junk characters do not affect the object and the mapping runs successfully.
 - Ensure that the folder path to the Amazon S3 bucket specified in the Amazon S3 V1 and V2 connection is the same. After the migration, if the folder path is different, or if the object in the S3 folder path is within a nested folder, the data preview fails but the mapping runs successfully.
To fix the data preview, reselect the source object in the mapping.

Partitioning

When you configure a passthrough partitioning for an Amazon S3 V1 mapping, the partitioning type is not retained and defaults to None after the migration. You need to manually configure partitioning after the migration.

Parameterization

Consider the following migration guidelines for parameterized mappings:

- If you create a completely parameterized mapping, do not specify the connection type in the connection parameterization. If you specify the connection type in the connection parameterization and then migrate the mapping, the mapping fails at runtime.
- When you parameterize both the connection and the objects in an Amazon S3 V1 mapping, the data preview in the mapping task might fail after the migration with the following error: `Internal error while processing request. Please contact Informatica Global Customer Support.`
To avoid this error, reselect the object in the mapping to preview the data.

Microsoft Azure Blob Storage

The following table lists the configured advanced source and target properties from Microsoft Azure Blob Storage V1 and V2 mappings that you can retain when you migrate to Microsoft Azure Blob Storage V3:

Transformations	Properties retained after migration
Source	<ul style="list-style-type: none">- Number of concurrent connections to Blob Store- Blob Name Override- Blob Container Override- Header in the first row of Blob
Target	<ul style="list-style-type: none">- Number of concurrent connections to Blob Store- Blob Name Override- Blob Container Override- Header in the first row of Blob- Compress newly created Blob- Write Strategy- Blob Type

Guidelines to migrate to Microsoft Azure Blob Storage V3

Some guidelines and restrictions apply to assets when you migrate the Microsoft Azure Blob Storage V1 and V2 connections to Microsoft Azure Blob Storage V3.

Guidelines applicable to Microsoft Azure Blob Storage V1 connection

The following guidelines are applicable when you migrate the Microsoft Azure Blob Storage V1 connection to Microsoft Azure Blob Storage V3.

Source transformation

Consider the following migration guidelines when you configure a read operation:

- After the migration, when you run a mapping to read data that contains double quotes ("), the double quotes is treated as a qualifier resulting in data mismatch in the target.
- After the migration, when you run a mapping to read data where a column name contains a single quote ('), the agent replaces the single quote in the metadata with an underscore (_) character resulting in failure.
- When you read data that contains special characters and create a target at runtime, data corruption occurs and the mapping fails. After the migration, data corruption does not occur and the same mapping runs successfully.
- When you read a compressed file either in the source object or in the overridden object, ensure that the file name has a .gz extension.
- The **Header in the first row of Blob** option in the source in a Microsoft Azure Blob Storage V1 mapping after the migration is mapped to the **Header Line** and **First Data Row** options in the Microsoft Azure Blob Storage V3 mapping.

If the **Header in the first row of Blob** option is enabled in a Microsoft Azure Blob Storage V1 mapping, after migration, the value **Header Line** is set to 1 and **First Data Row** is set to 2.

If the **Header in the first row of Blob** option is not enabled in a Microsoft Azure Blob Storage V1 mapping, after migration, the value **Header Line** is set to 0 and **First Data Row** is set to 1.

Target Transformation

Consider the following migration guidelines when you configure a write operation:

- After the migration, when you configure a Target transformation, the Success File Directory and Error File Directory target properties are not available in Microsoft Azure Blob Storage V3.
- The **Header in the first row of Blob** option in the target in a Microsoft Azure Blob Storage V1 mapping after the migration is mapped to the **Target Header** option in the Microsoft Azure Blob Storage V3 mapping.

If the **Header in the first row of Blob** option is enabled in a Microsoft Azure Blob Storage V1 mapping, after migration, the mapping writes the data with the header to the target.

If the **Header in the first row of Blob** option is not enabled in a Microsoft Azure Blob Storage V1 mapping, after migration, the mapping writes the data without the header to the target.

Connection

Before the migration, ensure that the container name in Microsoft Azure Blob Storage V1 and V3 connections is the same.

Guidelines applicable to Microsoft Azure Blob Storage V2 connection

The following guidelines are applicable when you migrate the Microsoft Azure Blob Storage V2 connection to Microsoft Azure Blob Storage V3.

Source transformation

Consider the following migration guidelines when you configure a read operation:

- If the override value for the blob container name in the Microsoft Azure Blob Storage V2 mapping ends with a forward slash (/), the mapping fails after the migration.
Before the migration, ensure that the override value for the blob container name does not end with forward slash (/).
- The **Header in the first row of Blob** option in the source in a Microsoft Azure Blob Storage V2 mapping after the migration is mapped to the **Header Line** and **First Data Row** options in the Microsoft Azure Blob Storage V3 mapping.

If the **Header in the first row of Blob** option is enabled in a Microsoft Azure Blob Storage V2 mapping, after migration, the value **Header Line** is set to 1 and **First Data Row** is set to 2.

If the **Header in the first row of Blob** option is not enabled in a Microsoft Azure Blob Storage V2 mapping, after migration, the value **Header Line** is set to 0 and **First Data Row** is set to 1.

Target transformation

Consider the following migration guidelines when you configure a write operation:

- If the override value for the blob container name in the Microsoft Azure Blob Storage V2 mapping ends with a forward slash (/), the mapping replaces the blob container with an unnamed folder in the target. After the migration, the same mapping replaces the blob container with the root directory in the target.
- The **Header in the first row of Blob** option in the target in a Microsoft Azure Blob Storage V2 mapping after the migration is mapped to the **Target Header** option in the Microsoft Azure Blob Storage V3 mapping.

If the **Header in the first row of Blob** option is enabled in a Microsoft Azure Blob Storage V2 mapping, after migration, the mapping writes the data with the header to the target.

If the **Header in the first row of Blob** option is not enabled in a Microsoft Azure Blob Storage V2 mapping, after migration, the mapping writes the data without the header to the target.

Compression

When you enable compression in a Microsoft Azure Blob Storage V2 mapping, the data is not compressed. After the migration, the same mapping uses Gzip compression to compress the data.

Guidelines applicable to Microsoft Azure Blob Storage V1 and V2 connections

The following guidelines are applicable when you migrate the Microsoft Azure Blob Storage V1 and V2 connections to Microsoft Azure Blob Storage V3.

Parameterization

Consider the following migration guidelines for parameterized Microsoft Azure Blob Storage mappings:

- If you create a completely parameterized mapping, do not specify the connection type in the connection parameterization. If you specify the connection type in the connection parameterization and then migrate the mapping, the mapping fails at runtime.
- When you parameterize both the connection and the objects in a Microsoft Azure Blob Storage V1 and V2 mapping, the data preview in the mapping task might fail after the migration with the following error:
`Internal error while processing request. Please contact Informatica Global Customer Support.`
To avoid this error, reselect the object in the mapping to preview the data.

Microsoft Azure Synapse SQL

The following table lists the configured advanced source, lookup, and target properties from Microsoft Azure SQL Data Warehouse V1 mappings that you can retain when you migrate to Microsoft Azure Synapse SQL:

Transformations	Properties retained after migration
Source	<ul style="list-style-type: none">- Azure Blob Container Name- Field Delimiter- Number of Concurrent Connections to Blob Store- Quote Character
Lookup	<ul style="list-style-type: none">- Azure Blob Container Name- Field Delimiter- Number of Concurrent Connections to Blob Store- Quote Character
Target	<ul style="list-style-type: none">- Azure Blob Container Name- Field Delimiter- Number of Concurrent Connections to Blob Store- Truncate Table- Quote Character- Forward Rejected Rows

The following table lists the configured advanced source, lookup, and target properties from Microsoft Azure SQL Data Warehouse V2 mappings that you can retain when you migrate to Microsoft Azure Synapse SQL:

Transformations	Properties retained after migration
Source	<ul style="list-style-type: none"> - Azure Blob Container Name - Field Delimiter - Number of Concurrent Connections to Blob Store - Pre SQL - Post SQL - SQL Override - On Pre-Post SQL Error - Quote Character
Lookup	<ul style="list-style-type: none"> - Azure Blob Container Name - Field Delimiter - Number of Concurrent Connections to Blob Store - Pre SQL - Post SQL - SQL Override - On Pre-Post SQL Error - Quote Character
Target	<ul style="list-style-type: none"> - Azure Blob Container Name - Field Delimiter - Number of Concurrent Connections to Blob Store - Truncate Table - Pre SQL - Post SQL - SQL Override - On Pre-Post SQL Error - Quote Character - Forward Rejected Rows

Guidelines to migrate to Microsoft Azure Synapse SQL

Some guidelines and restrictions apply to assets when you migrate the Microsoft Azure SQL Data Warehouse V1 and V2 connections to Microsoft Azure Synapse SQL.

Guidelines applicable to Microsoft Azure SQL Data Warehouse V1 connection

The following guidelines are applicable when you migrate the Microsoft Azure SQL Data Warehouse V1 connection to Microsoft Azure Synapse SQL.

Source transformation

Consider the following migration guidelines when you configure a read operation:

- After the migration, when you run a mapping to read data that contains special characters, the mapping fails with the following error: [FATAL] Exception occurred while validating record metadata: _____1 is invalid field name.
To avoid this error, after the migration, refresh the metadata or regenerate the fields and run the mapping again.
- After the migration, when you run a mapping to read data that contains unicode characters, the mapping fails with the following error: [ERROR] Invalid operation field name in CCI Metadata Client.
To avoid this error, after the migration, refresh the metadata or regenerate the fields and run the mapping again.

Target transformation

After the migration, when you configure a Target transformation, the Success File Directory and Error File Directory target properties are not available in Microsoft Azure Synapse SQL.

Guidelines applicable to Microsoft Azure SQL Data Warehouse V2 connection

When you configure any operation other than insert, and then migrate to Microsoft Azure Synapse SQL, you must select the applicable target attribute **Treat Source Row As** in the Microsoft Azure Synapse SQL target properties before you run the mapping.

Guidelines applicable to Microsoft Azure SQL Data Warehouse V1 and V2 connection

The following guidelines are applicable when you migrate the Microsoft Azure SQL Data Warehouse V1 and V2 connections to Microsoft Azure Synapse SQL.

Parameterization

Consider the following migration guidelines when you parameterize Microsoft Azure SQL Data Warehouse mappings:

- If you create a completely parameterized mapping, do not specify the connection type in the connection parameterization. If you specify the connection type in the connection parameterization and then perform the migration, the mapping fails at runtime.
- When you parameterize both the connection and the objects in a Microsoft Azure SQL Data Warehouse V1 and V2 mapping, the data preview in the mapping task might fail after the migration with the following error: `Internal error while processing request. Please contact Informatica Global Customer Support.`
To avoid this error, reselect the object in the mapping to preview the data.

Snowflake

The following table lists the configured advanced source, lookup, and target properties from Snowflake V1 mappings that you can retain when you migrate to Snowflake Data Cloud:

Transformations	Snowflake V1 properties retained after migration to Snowflake Data Cloud
Source	<ul style="list-style-type: none">- Database- Schema- Warehouse- Role- Pre SQL- Post SQL- Table name- SQL override- Tracing level
Lookup	<ul style="list-style-type: none">- Database- Schema- Warehouse- Role
Target	<ul style="list-style-type: none">- Database- Schema- Warehouse- Role- Pre SQL- Post SQL- Batch row size- Number of local staging files- Truncate target table- Additional runtime properties- Table name- Rejected file path- Success file directory- Error file directory- Forward rejected rows

Consider the following rules before you migrate the Snowflake V1 connection referenced in an existing mapping to the Snowflake Data Cloud connection:

- If the existing mapping contains multiple objects, the relationship between these objects is not preserved after you switch to the new connection.

Troubleshooting the migration

Consider the following troubleshooting tips when you switch the old connection to the current connection in assets:

No meter exists and returns 400 bad request error

If your organization is not enabled for metering, the migration API response returns the following error:

```
{
  "code": "0",
  "description": "400 Bad Request: \\{\\error\\":{\\code
\\":\\\"V3API_ContainerError_010\\\",\\message\\":
\\\"No Meter_Exists\\\",\\debugMessage\\\":\\\"No Meter Exists\\\",\\requestId\\":
\\\"fjntYutce6k1H8pfBXqyh\\\",\\details\\\":null}}\\\""
```

```
    "statusCode": 400
  }
```

In the Administrator service, ensure that metering is enabled for the **Number of projects** and **Number of folders** under platform as the service.

Mapping is not valid and returns 500 internal server error

If the connection that you want to migrate is referenced with any mapping that is not valid, the migration fails with the following error:

```
{
  "code": "0",
  "description": "500 Internal Server Error: \\{\\\"error\\\":{\\\"code
\\\":\\\"MigrationSvc_031\\\",\\\"message\\\":\\\"Import failed\\\",
\\\"requestId\\\":null,\\\"details\\\":{\\\"message\\\":\\\"Object name [multi_src_tgt_param] type [MTT]
path [/Default]
message [The mapping 'multi_src_tgt_param' cannot be deployed as it is not valid.]}\\\",
{\\\"message\\\":\\\"Object name [multi_src_tgt_param] type [MTT] path [/RS_CTK_auto]
message [Object skipped because of previous failures]}\\\",
{\\\"message\\\":\\\"Object name [rs_ckpt_inout_param] type [MTT] path [/RS_CTK_auto]
message [Object skipped because of previous failures]}\\\",
{\\\"message\\\":\\\"Object name [rs_ckpt_input_param] type [MTT] path [/RS_CTK_auto] message
[Object skipped because of previous failures]}\\\"}}\\\",
  \"statusCode\": 400
}
```

Author

Connectivity Engineering Team