Informatica®

Informatica® Cloud Data Integration

Microsoft Azure Data Lake Storage Gen2 Connector

Informatica Cloud Data Integration Microsoft Azure Data Lake Storage Gen2 Connector
April 2024

Publication Date: 2024-04-24

Table of Contents

Preface 5

Informatica Resources 5

- Informatica Documentation 5
- Informatica Intelligent Cloud Services web site 5
- Informatica Intelligent Cloud Services Communities 5
- Informatica Intelligent Cloud Services Marketplace 5
- Data Integration connector documentation 6
- Informatica Knowledge Base 6
- Informatica Intelligent Cloud Services Trust Center 6
- Informatica Global Customer Support 6

Chapter 1: Introduction to Microsoft Azure Data Lake Storage Gen2 Connector 7

Microsoft Azure Data Lake Storage Gen2 Connector example 7

Microsoft Azure Data Lake Storage Gen2 Connector assets 8

File formats 8

Chapter 2: Connections for Microsoft Azure Data Lake Storage Gen2 9

Prepare for authentication 9

- Managed identity authentication 10

Connect to Microsoft Azure Data Lake Storage Gen2 10

- Before you begin 10
- Connection details 10
- Authentication types 11

Proxy server settings 13

- Bypass the proxy server 14

Chapter 3: Mappings for Microsoft Azure Data Lake Storage Gen2 15

Microsoft Azure Data Lake Storage Gen2 sources in mappings 15

- Directory source in Microsoft Azure Data Lake Storage Gen2 sources 18
- Wildcard characters 19
- Reading files from subdirectories 20
- Incrementally loading files 20
- SQL ELT optimization 21

Microsoft Azure Data Lake Storage Gen2 targets in mappings 21

- Specifying a target 24
- Target time stamps 25
- Target partitioning 25

File formatting options 26

Fixed-width file formats 28

FileName field. . . . 29
Reading source objects path. . . . 29
Writing to target objects. . . . 30
Rules and guidelines for FileName field. . . . 31
Directory-level partitioning. . . . 32
Rules and guidelines for reading from and writing to a partition folder. . . . 35
Parameterization. . . . 36
Mappings in advanced mode example. . . . 37
Rules and guidelines for mappings. . . . 38

Chapter 4: Migrating a mapping. . . . 41

Use the same object path for the migrated mapping. . . . 41
Use a different object path for the migrated mapping. . . . 41
Migration options. . . . 42
General rules and guidelines. . . . 43

Chapter 5: Data type reference 44

Flat file data types and transformation data types. . . . 44
Avro data types and transformation data types. . . . 45
JSON data types and transformation data types. . . . 46
ORC data types and transformation data types. . . . 47
Parquet data types and transformation data types. . . . 48

Chapter 6: Troubleshooting. . . . 50

Troubleshooting a mapping. . . . 50
Troubleshooting a mapping in advanced mode. . . . 52

Index. . . . 53

Preface

Use *Microsoft Azure Data Lake Storage Gen2 Connector* to learn how to read from or write to Microsoft Azure Data Lake Storage Gen2. Learn to create a connection, develop and run mappings, mapping tasks, dynamic mapping tasks, and data transfer tasks in Cloud Data Integration.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to Microsoft Azure Data Lake Storage Gen2 Connector

You can use Microsoft Azure Data Lake Storage Gen2 Connector to securely read data from or write to Microsoft Azure Data Lake Storage Gen2.

Use Microsoft Azure Data Lake Storage Gen2 Connector to read and write flat files and complex files such as Avro, JSON, ORC, and Parquet. You can use Microsoft Azure Data Lake Storage Gen2 objects as sources and targets in mappings and mapping tasks.

You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality. A mapping in advanced mode can run on an advanced cluster hosted on Microsoft Azure, local cluster, or a self-service cluster.

Microsoft Azure Data Lake Storage Gen2 Connector example

You work as a data analyst for a large financial enterprise. The enterprise performs risk management, fraud detection, and other analysis with Azure Data Lake Analytics. You need to write the delimited data to Microsoft Azure Data Lake Storage Gen2 to perform the analytics.

You can use Microsoft Azure Data Lake Storage Gen2 Connector and create a mapping task to read data from sources such as relational or transactional database or other applications such as Salesforce and write data to Microsoft Azure Data Lake Storage Gen2. After the data is available in the Microsoft Azure Data Lake Storage Gen2, you can perform the data analytics.

Microsoft Azure Data Lake Storage Gen2 Connector assets

Create assets in Data Integration to integrate data using Microsoft Azure Data Lake Storage Gen2 Connector.

When you use Microsoft Azure Data Lake Storage Gen2 Connector, you can include the following Data Integration assets:

- Data transfer task
- Dynamic mapping task
- Mapping
- Mapping task

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

File formats

The following table lists the format types that Microsoft Azure Data Lake Storage Gen2 Connector uses to read and write data in mappings and mappings in advanced mode:

Mappings	Mappings in advanced mode
Flat Binary Avro (Primitive data types) JSON (Primitive data types) ORC (Primitive data types) Parquet (Primitive data types)	Flat Avro (Primitive and hierarchical data types) JSON (Primitive and hierarchical data types) ORC (Primitive and hierarchical data types) Parquet (Primitive and hierarchical data types) Discover structure*

*You can only read data of the discover structure file format with mappings in advanced mode.

CHAPTER 2

Connections for Microsoft Azure Data Lake Storage Gen2

Create a Microsoft Azure Data Lake Storage Gen2 connection to securely read data from or write data to Microsoft Azure Data Lake Storage Gen2.

You can use a Microsoft Azure Data Lake Storage Gen2 connection to specify sources and targets in mappings and mapping tasks.

Prepare for authentication

You can configure Shared Key, Managed Identity, and Service Principal authentication types to access Microsoft Azure Data Lake Storage Gen2.

Before you configure the authentication, you must create a storage account to use with Microsoft Azure Data Lake Storage Gen2 and create a blob container in the storage account. You can use role-based access control or access control lists to authorize the users to access the resources in the storage account.

You must also register an application in Azure Active Directory to authenticate users to access the Microsoft Azure Data Lake Storage Gen2 account. You can use role-based access control or access control lists to authorize the application.

You must also create an Azure Active Directory web application for service-to-service authentication with Microsoft Azure Data Lake Storage Gen2 and ensure that you have superuser privileges to access the folders or files created in the application.

For more information about these prerequisite tasks, see the Informatica How-To Library article, Prerequisites to create a Microsoft Azure Data Lake Storage Gen2 connection.

After you complete the prerequisite tasks, you need to keep the authentication details handy based on the authentication type that you want to use:

- To use service principal authentication, you need the client ID, client secret, and tenant ID for your application registered in the Azure Active Directory.
- To use shared key authentication, you need the account key for the Microsoft Azure Data Lake Storage Gen2 account.
- To use managed identity authentication, you need the client ID or application ID for your application registered in the Azure Active Directory. Before you get the client ID or application ID, be sure to complete certain prerequisites.

Managed identity authentication

Managed Identity authentication uses managed identities in Azure Active Directory to authenticate and authorize access to Azure resources securely.

Before you use managed identity authentication to connect to Microsoft Azure Data Lake Storage Gen2, be sure to complete certain prerequisites.

1. Create an Azure virtual machine.
2. Install the Secure Agent on the Azure virtual machine.
3. Enable system assigned identity or user assigned identity for the Azure virtual machine.
   If you enable both and do not specify the client ID, the system assigned identity is used for authentication.
4. After you add or remove a managed identity, restart the Azure virtual machine.

Connect to Microsoft Azure Data Lake Storage Gen2

Let's configure the Microsoft Azure Data Lake Storage Gen2 connection properties to connect to Microsoft Azure Data Lake Storage Gen2.

Before you begin

Before you get started, you'll need to get information from your Microsoft Azure Data Lake Storage Gen2 account based on the authentication type that you want to configure.

Check out "Prepare for authentication" on page 9 to learn more about the authentication prerequisites.

Connection details

The following table describes the basic connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.
Type	Microsoft Azure Data Lake Storage Gen2

Property	Description
Use Secret Vault	Stores sensitive credentials for this connection in the secrets manager that is configured for your organization. This property appears only if secrets manager is set up for your organization. When you enable the secret vault in the connection, you can select which credentials that the Secure Agent retrieves from the secrets manager. If you don't enable this option, the credentials are stored in the repository or on a local Secure Agent, depending on how your organization is configured. For information about how to configure and use a secrets manager, see "Secrets manager configuration" in the Administrator help.
Runtime Environment	The name of the runtime environment where you want to run tasks. Select a Secure Agent, Hosted Agent, or serverless runtime environment. Do not use a Hosted Agent if you use the connection in mappings in advanced mode.
AccountName	Microsoft Azure Data Lake Storage Gen2 account name or the service name.

Authentication types

You can select service principal authentication, shared key authentication, and managed identity authentication to access the Microsoft Azure Data Lake Storage Gen2 account.

Select your preferred authentication type and then configure the authentication-specific parameters.

Service principal authentication

Service principal authentication uses the client ID, client secret, and tenant ID to connect to Microsoft Azure Data Lake Storage Gen2.

The following table describes the basic connection properties for service principal authentication:

Property	Description
Client ID	The client ID of your application. Specify the client ID for your application registered in the Azure Active Directory.
Client Secret	The client secret key generated for the client ID. Specify the client secret key to complete the OAuth authentication in the Azure Active Directory.
Tenant ID	The directory ID of the Azure Active Directory.
File System Name	The name of the file system in the Microsoft Azure Data Lake Storage Gen2 account.

Property	Description
Directory Path	The path of a directory without the file system name. You can select from the following directory structures: - / for root directory - /dir1 - dir1/dir2 Default is /.
Adls Gen2 Endpoint	The type of Microsoft Azure endpoints. Select one of the following endpoints: - core.windows.net. Connects to Azure endpoints. - core.usgovcloudapi.net. Connects to US government Microsoft Azure Data Lake storage Gen2 endpoints. - core.chinacloudapi.cn. Connects to Microsoft Azure Data Lake storage Gen2 endpoints in the China region. Default is core.windows.net. **Note:** You cannot configure the Azure Government endpoints for mappings in advanced mode.

Shared key authentication

Shared key authentication uses the account key to connect to Microsoft Azure Data Lake Storage Gen2.

The following table describes the basic connection properties for shared key authentication:

Property	Description
Account Key	The account key for the Microsoft Azure Data Lake Storage Gen2 account.
File System Name	The name of the file system in the Microsoft Azure Data Lake Storage Gen2 account.
Directory Path	The path of a directory without the file system name. You can select from the following directory structures: - / for root directory - /dir1 - dir1/dir2 Default is /.
Adls Gen2 Endpoint	The type of Microsoft Azure endpoints. Select one of the following endpoints: - core.windows.net. Connects to Azure endpoints. - core.usgovcloudapi.net. Connects to US government Microsoft Azure Data Lake storage Gen2 endpoints. - core.chinacloudapi.cn. Connects to Microsoft Azure Data Lake storage Gen2 endpoints in the China region. Default is core.windows.net. **Note:** You cannot configure the Azure Government endpoints for mappings in advanced mode.

Managed identity authentication

Managed identity authentication authenticates using identities that are assigned to applications in Azure to access Azure resources in Microsoft Azure Data Lake Storage Gen2.

When you create a Microsoft Azure Data Lake Storage Gen2 connection, select the Azure virtual machine on which you have installed the Secure Agent. If you enable system assigned identity, assign the required role or permissions to the Azure virtual machine to run the mappings and tasks. If you enable user assigned identity, assign the required role or permissions to the user assigned identity. For example, if you use role-based access control, assign the Storage Blob Data Contributor role and if you use access control lists, assign the read, write, and execute permissions.

The following table describes the basic connection properties for managed identity authentication:

Property	Description
Client ID	The client ID of your application. To use managed identity authentication, specify the client ID for the user-assigned managed identity. Leave the field blank in the following scenarios: - If the permission is provided by system-assigned managed identity. - If there is no system-assigned identity but only a single user-assigned managed identity.
File System Name	The name of the file system in the Microsoft Azure Data Lake Storage Gen2 account.
Directory Path	The path of a directory without the file system name. You can select from the following directory structures: - `/` for root directory - `/dir1` - `dir1/dir2` Default is `/`.
Adls Gen2 End-point	The type of Microsoft Azure endpoints. Select one of the following endpoints: - core.windows.net. Connects to Azure endpoints. - core.usgovcloudapi.net. Connects to US government Microsoft Azure Data Lake storage Gen2 endpoints. - core.chinacloudapi.cn. Connects to Microsoft Azure Data Lake storage Gen2 endpoints in the China region. Default is core.windows.net. **Note:** You cannot configure the Azure Government endpoints for mappings in advanced mode.

Proxy server settings

If your organization uses an outgoing proxy server to connect to the Internet, the Secure Agent connects to Informatica Intelligent Cloud Services through the proxy server.

You can configure the Secure Agent to use the proxy server on Windows and Linux. You can use the unauthenticated or authenticated proxy server.

Note: You cannot use a proxy server with managed identity authentication.

You can use one of the following types of proxy servers:

- Unauthenticated proxy - Requires only the host and port address for configuration.
- Authenticated proxy - Requires the host address, port address, user name, and password for configuration.

To configure proxy settings for the Secure Agent, use one of the following methods:

- Configure the Secure Agent through the Secure Agent Manager on Windows or shell command on Linux.

  For instructions, see "Configure the proxy settings on Windows" or "Configure the proxy settings on Linux" in *Getting Started* in the Data Integration help .
- Configure the JVM options for the DTM in the Secure Agent properties. For instructions, see the Proxy server settings Knowledge Base article.

To configure proxy settings for the serverless runtime environment, see "Using a proxy server" in *Runtime Environments* in the Administrator help.

Bypass the proxy server

You can bypass the proxy server settings configured for the Secure Agent.

Perform the following steps to bypass the proxy server:

1. Navigate to the following directory:
   `<Secure Agent installation directory>/apps/agentcore`
2. Specify the following command in the `proxy.ini` file:
   `InfaAgent.NonProxyHost=localhost|{*}core.windows.net|127.|[\:\:1]*`

   To bypass proxy server for service principal authentication, append `login.microsoftonline.com` to the command.

   To bypass proxy server for managed identity authentication, append `169.254.169.254` to the command.

   For example,

   ```
   InfaAgent.NonProxyHost=localhost|127.*|[\:\:1]|<accountname>.blob.core.windows.net|
   <accountname>.dfs.core.windows.net|<accountname>.blob.core.windows.net|
   login.microsoftonline.com|169.254.169.254
   ```
3. Restart the Secure Agent.

CHAPTER 3

Mappings for Microsoft Azure Data Lake Storage Gen2

When you configure a mapping, you describe the flow of data from the source to the target.

A mapping defines reusable data flow logic that you can use in mapping tasks.

When you create a mapping, you define the Source transformation and Target transformation to represent a Microsoft Azure Data Lake Storage Gen2 object. Use the Mapping Designer in Data Integration to add the Source or Target transformations in the mapping canvas and configure the Microsoft Azure Data Lake Storage Gen2 source and target properties.

In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

You can use Monitor to monitor the jobs.

Microsoft Azure Data Lake Storage Gen2 sources in mappings

In a mapping, you can configure a source transformation to represent a single Microsoft Azure Data Lake Storage Gen2 object.

The following table describes the Microsoft Azure Data Lake Storage Gen2 source properties that you can configure in a source transformation:

Property	Description
Connection	Name of the source connection. Select a source connection or click **New Parameter** to define a new parameter for the source connection. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. Ensure that the parameter file is in the correct format. When you switch between a non-parameterized and a parameterized Microsoft Azure Data Lake Storage Gen2 connection, the advanced property values are retained.
Source Type	Select Single Object or Parameter.

Property	Description
Object	Name of the source object. Ensure that the headers or file data does not contain special characters.
Parameter	Select an existing parameter for the source object or click **New Parameter** to define a new parameter for the source object. The **Parameter** property appears only if you select Parameter as the source type. When you parameterize the source object, specify the complete object path including the file system in the default value of the parameter. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. Ensure that the parameter file is in the correct format.
Format	Specifies the file format that the Microsoft Azure Data Lake Storage Gen2 Connector uses to read data from Microsoft Azure Data Lake Storage Gen2. You can select the following file format types: - Flat - Avro - Parquet - JSON - ORC - Discover Structure[1] Default is **None**. If you select **None** as the format type, Microsoft Azure Data Lake Storage Gen2 Connector reads data from Microsoft Azure Data Lake Storage Gen2 files in binary format. You cannot read a JSON file that exceeds 1 GB. **Note:** Ensure that the source file is not empty. For more information, see "File formatting options" on page 26
Intelligent Structure Model[1]	Applies to Discover Structure format type. Determines the underlying patterns in a sample file and auto-generates a model for files with the same data and structure. Select one of the following options to associate a model with the transformation: - Select. Select an existing model. - New. Create a new model. Select **Design New** to create the model. Select **Auto-generate from sample file for Intelligent Structure Discovery** to generate a model based on sample input that you select. Select one of the following options to validate the XML source object against an XML-based hierarchical schema: - Source object doesn't require validation. - Source object requires validation against a hierarchical schema. Select to validate the XML source object against an existing or a new hierarchical schema. When you create a mapping task, on the **Runtime Options** tab, you configure how Data Integration handles the schema mismatch. You can choose to skip the mismatched files and continue to run the task or stop the task when the task encounters the first file that does not match. For more information, see *Components*.

[1]Applies only to mappings in advanced mode.

The following table describes the Microsoft Azure Data Lake Storage Gen2 source advance properties:

Property	Description
Concurrent Threads[1]	Number of concurrent connections to extract data from the Microsoft Azure Data Lake Storage Gen2. When reading a large file or object, you can spawn multiple threads to process data. Configure **Block Size** to divide a large file into smaller parts. Default is 4. Maximum is 10.
Filesystem Name Override	Overrides the default file system name.
Source Type	Select the type of source from which you want to read data. You can select the following source types: - File - Directory Default is File.
Allow Wildcard Characters	Indicates whether you want to use wildcard characters for the directory source type. For more information, see "Wildcard characters" on page 19.
Directory Override	Microsoft Azure Data Lake Storage Gen2 directory that you use to read data. Default is root directory. The directory path specified at run time overrides the path specified while creating a connection. You can specify an absolute or a relative directory path: - Absolute path - The Secure Agent searches this directory path in the specified file system. Example of absolute path: `Dir1/Dir2` - Relative path - The Secure Agent searches this directory path in the native directory path of the object. Example of relative path: `/Dir1/Dir2` When you use the relative path, the imported object path is added to the file path used during the metadata fetch at runtime. Do not specify a root directory (`/`) to override the directory.
File Name Override	Source object. Select the file from which you want to read data. The file specified at run time overrides the file specified in Object.
Block Size[1]	Applicable to flat file format. Divides a large file into smaller specified block size. When you read a large file, divide the file into smaller parts and configure concurrent connections to spawn the required number of threads to process data in parallel. Specify an integer value for the block size. Default value in bytes is 8388608.
Timeout Interval	Not applicable.
Recursive Directory Read	Indicates whether you want to read objects stored in subdirectories in mappings. For more information, see "Reading files from subdirectories" on page 20
Incremental File Load[2]	Indicates whether you want to incrementally load files when you use a directory as the source for mappings in advanced mode. When you incrementally load files, the mapping task reads and processes only files in the directory that have changed since the mapping task last ran. For more information, see "Incrementally loading files" on page 20.

Property	Description
Compression Format	Reads compressed data from the source. Select one of the following options: - None. Select to read Avro, ORC, and Parquet files that use Snappy compression. The compressed files must have the `.snappy` extension. You cannot read compressed JSON files. - Gzip. Select to read flat files and Parquet files that use Gzip compression. The compressed files must have the `.gz` extension. You cannot preview data for a compressed flat file.
Interim Directory[1]	Optional. Applicable to flat files and JSON files. Path to the staging directory in the Secure Agent machine. Specify the staging directory where you want to stage the files when you read data from Microsoft Azure Data Lake Storage Gen2. Ensure that the directory has sufficient space and you have write permissions to the directory. Default staging directory is `/tmp`. You cannot specify an interim directory when you use the Hosted Agent.
Tracing Level	Sets the amount of detail that appears in the log file. You can choose terse, normal, verbose initialization or verbose data. Default is normal.

[1]Doesn't apply to mappings in advanced mode.
[2]Applies only to mappings in advanced mode.

Directory source in Microsoft Azure Data Lake Storage Gen2 sources

You can select the type of source from which you want to read data.

You can select the following type of sources from the **Source Type** option under the advanced source properties:

- File
- Directory

Use the following rules and guidelines to select **Directory** as the source type:

- All the source files in the directory must contain the same metadata.
- All the files must have data in the same format. For example, delimiters, header fields, and escape characters must be same.
- All the files under a specified directory are parsed. To parse the files in the subdirectories, use recursive read.
  For more information, see Reading files from subdirectories.
- When you run a mapping that reads data from a directory, the agent creates a single file in the target. When you create a mapping in advanced mode, the agent creates multiple files in the target.

Wildcard characters

When you read data from an Avro, flat, JSON, ORC, or Parquet file, you can use wildcard characters to specify the source file name.

To use wildcard characters for the source file name, select the source type as **Directory** and enable the **Allow Wildcard Characters** option in the advanced source properties.

When you read an Avro, JSON, ORC, Parquet, or flat file, you can use the `?` and `*` wildcard characters to define one or more characters in a search.

You can use the following wildcard characters:

? (Question mark)

The question mark character (?) allows one occurrence of any character. For example, if you enter the source file name as `a?b.txt`, the Secure Agent reads data from files with the following names:

- `a1b.txt`
- `a2b.txt`
- `aab.txt`
- `acb.txt`

*** (Asterisk)**

The asterisk mark character (*) allows zero or more than one occurrence of any character. If you enter the source file name as `a*b.txt`, the Secure Agent reads data from files with the following names:

- `aab.txt`
- `a1b.txt`
- `ab.txt`
- `abc11b.txt`

Rules and guidelines for wildcard characters

Consider the following rules and guidelines when you use wildcard characters:

Mappings

- You cannot use wildcard characters when you read from partition columns.
- When you read a complex file in a mapping, do not use a tilde (~) in the sub-directory name or file name.
- When you use wildcard characters in directory override, the Secure Agent reads data from the folders as well as the files that match the name pattern.

Mappings in advanced mode

- When you read a flat file or complex file and enable wildcard characters, ensure that the path specified in the directory override or file name override matches the file path in the source.
- When you use wildcard characters, ensure that the file name does not start with a special character.
- When you read a flat file, do not use the following special characters in the directory name or sub-directory name in the directory override:

  `[] {} " ' + ^ % * ? space`

Reading files from subdirectories

You can read objects stored in subdirectories in Microsoft Azure Data Lake Storage Gen2 in mappings.

You can use recursive read for flat files and complex files in mappings. When you create a mapping in advanced mode, you cannot use recursive read for flat files.

To enable recursive read, select the source type as **Directory** in the advanced source properties. Enable the **Recursive Directory Read** advanced source property to read objects stored in subdirectories.

Rules and guidelines for reading from subdirectories

Consider the following rules and guidelines when you read objects stored in subdirectories:

Mappings

- When you read from or write to a flat file in Microsoft Azure Data Lake Storage Gen2, ensure that the directory or subdirectory name does not contain the percentage (%) character. Else, the mapping fails.
- You cannot use recursive read when you read from partition columns.
- When you read a complex file in a mapping, do not use a tilde (~) in the subdirectory name or file name.
- When the FileName field for the source and target is mapped, the file is created in the following format:
  `target_filename=source_directory_subdirectory1_sudirectory2`
- When you read a flat file with only headers and no data and map the FileName field, the expected directory structure is not created with the FileName field.

Mappings in advanced mode

- When you read a complex file and enable recursive read, ensure that the path specified in the directory override or file name override matches the file path in the source.
- When you read a flat file, do not use the following special characters in the directory name or sub-directory name in the directory override:
  `[] {} " ' + ^ % * ? space`

Incrementally loading files

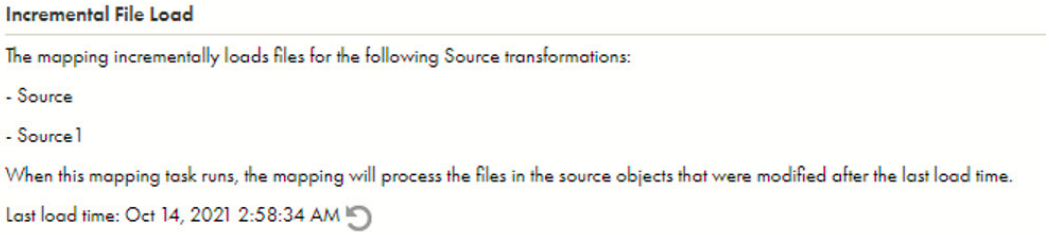
You can incrementally load source files in a directory to read and process only the files that have changed since the last time the mapping task ran.

You can incrementally load files only from mappings in advanced mode. Ensure that all of the source files exist in the same Cloud environment.

To incrementally load source files, select **Incremental File Load** and **Directory** as the source type in the advanced read options of the Microsoft Azure Data Lake Storage Gen2 data object.

When you incrementally load files from Microsoft Azure Data Lake Storage Gen2, the job loads files that have changed from the last load time to five minutes before the job started running. For example, if you run a job at 2:00 p.m, the job loads files changed before 1:55 p.m. The five-minute buffer ensures that the job loads only complete files because uploading objects on Microsoft Azure Data Lake Storage Gen2 can take a few minutes to complete.

When you configure a mapping task, the **Incremental File Load** section lists the Source transformations that will incrementally load files and the time that the last job completed loading the files. By default, the next job that runs checks for files modified after the last load time.



You can also override the load time that the mapping uses to look for changed files in the specified source directory. You can reset the incremental file load settings to perform a full load of all the changed files in the directory, or you can configure a time that the mapping uses to look for changed files.

A mapping in advanced mode that incrementally loads a directory that contains complex file formats such as Parquet and Avro fails if there are no new or changed files in the source since the last run.

For more information on incremental loading, see Reprocessing incrementally-loaded source files in *Tasks* in the Data Integration documentation.

SQL ELT optimization

You can enable full SQL ELT optimization when you want to load data from Microsoft Azure Data Lake Storage Gen2 sources to your data warehouse in Microsoft Azure Synapse SQL. While loading the data to Microsoft Azure Synapse SQL, you can transform the data as per your data warehouse model and requirements. When you enable full SQL ELT optimization on a mapping task, the mapping logic is pushed to the Azure environment to leverage Azure commands. For more information, see the help for Microsoft Azure Synapse SQL Connector.

If you need to load data to any other supported cloud data warehouse, see the connector help for the applicable cloud data warehouse.

Microsoft Azure Data Lake Storage Gen2 targets in mappings

In a mapping, you can use a Microsoft Azure Data Lake Storage Gen2 object as a target.

When you use Microsoft Azure Data Lake Storage Gen2 target objects, you can select a Microsoft Azure Data Lake Storage Gen2 Gen2 collection as target. You can configure Microsoft Azure Data Lake Storage Gen2 target properties on the Target page of the Mapping wizard. When you write data to Microsoft Azure Data Lake Storage Gen2, you can use the create target field to create a target at run time. When you create a new target based on the source, you must remove all the binary fields from the field mapping.

The following table describes the Microsoft Azure Data Lake Storage Gen2 target properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the target connection. Select a target connection or click **New Parameter** to define a new parameter for the target connection. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. When you switch between a non-parameterized and a parameterized Microsoft Azure Data Lake Storage Gen2 connection, the advanced property values are retained.
Target Type	Select Single Object or Parameter.
Object	Name of the target object. You can select an existing object or create a new target at runtime. When you select **Create New at Runtime**, enter a name for the target object and select the source fields that you want to use. By default, all source fields are used. The target name can contain alphanumeric characters. You can use only a period (.), an underscore (_), an at the rate sign (@), a dollar sign ($), and a percentage sign (%) special characters in the file name. Ensure that the headers or file data does not contain special characters. You can use parameters defined in a parameter file in the target name. When you select the Create Target option, you cannot parameterize the target at runtime. **Note:** When you write data to a flat file created at runtime, the target flat file contains a blank line at the end of the file.
Parameter	Select an existing parameter for the target object or click **New Parameter** to define a new parameter for the target object. The **Parameter** property appears only if you select Parameter as the target type. When you parameterize the target object, specify the complete object path including the file system in the default value of the parameter. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. Ensure that the parameter file is in the correct format.
Format	Specifies the file format that the Microsoft Azure Data Lake Storage Gen2 Connector uses to write data to Microsoft Azure Data Lake Storage Gen2. You can select the following file format types: - Flat - Avro - Parquet - JSON - ORC Default is **None**. If you select **None** as the format type, Microsoft Azure Data Lake Storage Gen2 Connector writes data to Microsoft Azure Data Lake Storage Gen2 files in binary format. For more information, see "File formatting options" on page 26
Operation	The target operation. Select Insert to insert data to a Microsoft Azure Data Lake Storage Gen2 target.

Note: When you use the **Create Target** option and specify an object name with extension that does not match the `Format Type` under **Formatting Options**, the Secure Agent ignores the format type you specified under **Formatting Options**.

For example, if you select `Parquet` format type and specify `customer.avro` in the object name in the **Target Object** dialog box, the Secure Agent ignores Parquet and creates an Avro target file.

The following table describes the advanced target properties for Microsoft Azure Data Lake Storage Gen2:

Advanced Target Property	Description
Concurrent Threads[1]	Number of concurrent connections to load data from the Microsoft Azure Data Lake Storage Gen2. When writing a large file, you can spawn multiple threads to process data. Configure **Block Size** to divide a large file into smaller parts. Default is 4. Maximum is 10.
Filesystem Name Override	Overrides the default file name.
Directory Override	Microsoft Azure Data Lake Storage Gen2 directory that you use to write data. Default is root directory. The Secure Agent creates the directory if it does not exist. The directory path specified at run time overrides the path specified while creating a connection. You can specify an absolute or a relative directory path: - Absolute path - The Secure Agent searches this directory path in the specified file system. Example of absolute path: `Dir1/Dir2` - Relative path - The Secure Agent searches this directory path in the native directory path of the object. Example of relative path: `/Dir1/Dir2` When you use the relative path, the imported object path is added to the file path used during the metadata fetch at runtime. Do not specify a root directory (`/`) to override the directory.
File Name Override	Target object. Select the file from which you want to write data. The file specified at run time overrides the file specified in Object.
Write Strategy	Applicable to complex and flat files. When you create a mapping, you can use the overwrite and append write strategy for flat files. However, you can use only the overwrite strategy for complex files. When you create a mapping in advanced mode, you can use the overwrite and append write strategy for both flat files and complex files. When you create a new target at runtime and use the append strategy, the mapping creates a new target file and writes the data to the file. The mapping appends data in subsequent runs. When you append data for mappings in advanced mode, the data is appended as a new part file in the existing target directory. The maximum size of data that you can append is 450 MB. Default is overwrite.
Block Size[1]	Applicable to flat, Avro, and Parquet file formats. Divides a large file into smaller specified block size. When you write a large file, divide the file into smaller parts and configure concurrent connections to spawn the required number of threads to process data in parallel. Specify an integer value for the block size. Default value in bytes is 8388608.
Compression Format	Compresses and writes data to the target based on the format you specify. Select one of the following options: - **None**. Select to write Avro, ORC, and Parquet files that use Snappy compression. You cannot write compressed JSON files. - **Gzip**. Select to write flat files and Parquet files that use Gzip compression. When the task runs, the file extensions `.gz` or `.snappy` do not appear in target object name.

Advanced Target Property	Description
Timeout Interval	Not applicable.
Interim Directory[1]	Optional. Applicable to flat files and JSON files. Path to the staging directory in the Secure Agent machine. Specify the staging directory where you want to stage the files when you write data to Microsoft Azure Data Lake Storage Gen2. Ensure that the directory has sufficient space and you have write permissions to the directory. Default staging directory is `/tmp`. You cannot specify an interim directory for mappings in advanced mode. You cannot specify an interim directory when you use the Hosted Agent.
Forward Rejected Rows[1]	Configure the transformation to either pass rejected rows to the next transformation or drop them.
[1]Doesn't apply to mappings in advanced mode.	

Specifying a target

You can use an existing target or create a target to hold the results of a mapping. If you choose to create the target, the agent creates the target when you run the task.

To specify the target properties, follow these steps:

1. Select the Target transformation in the mapping.
2. On the **Incoming Fields** tab, configure field rules to specify the fields to include in the target.
3. To specify the target, click the **Target** tab.
4. Select the target connection.
5. For the target type, choose **Single Object** or **Parameter**.
6. Specify the target object or parameter.
   - To create a target file at run time, enter the name for the target file including the extension, for example, `Accounts.csv`.

     Note: When you read from a flat file, ensure that the file contains some data and not the header alone. If the file has only a header, the header is not written to the target.
   - If you want the file name to include a time stamp, click **Handle Special Characters** and add special characters to the file name. For example, add the special characters shown here to include all the time stamp information: `Accounts_%d%m%y%T.csv`.
   - If you want the folder name to include a time stamp, click **Handle Special Characters** and add the folder name separated with a back slash (\) followed by the file name. For example, `%Y%m%d \Target_filename_%m.csv`.

   Note: The Handle Special Characters option is not applicable to mappings in advanced mode.
7. Click **Formatting Options** if you want to configure the formatting options for the file, and click **OK**.
8. Click **Select** and choose a target object. You can select an existing target object or create a new target object at run time and specify the object name.
9. Specify Advanced properties for the target, if needed.

Target time stamps

When you create a target at run time in a mapping, you can append time stamp information to the file name to show when the file is created.

When you specify the file name for the target file, include special characters based on Linux STRFTIME function formats that the mapping task uses to include time stamp information in the file name. The time stamp is based on the organization's time zone.

You cannot append time stamp information to the file name for mappings in advanced mode.

The following table describes some common STRFTIME function formats that you might use in a mapping or mapping task:

Special Character	Description
%d	Day as a two-decimal number, with a range of 01-31.
%m	Month as a two-decimal number, with a range of 01-12.
%y	Year as a two-decimal number without the century, with range of 00-99.
%Y	Year including the century, for example 2015.
%T	Applicable only to flat files. Time in 24-hour notation, equivalent to %H:%:M:%S.
%H	Hour in 24-hour clock notation, with a range of 00-24.
%I	Hour in 12-hour clock notation, with a range of 01-12.
%M	Minute as a decimal, with a range of 00-59.
%S	Second as a decimal, with a range of 00-60.
%p	Either AM or PM.

Note: For complex files, instead of %T you can use the equivalent %H_%M_%S.

Target partitioning

You can configure partitioning to optimize the mapping performance at run time when you write data to Microsoft Azure Data Lake Storage Gen2. You can configure target partitioning only in mappings.

The partition type controls how the agent distributes data among partitions at partition points. With partitioning, the Secure Agent distributes rows of target data based on the number of threads that you define as partition.

For example, if there are three partitions in the source, the Secure Agent writes separate files for each partition in the Microsoft Azure Data Lake Storage Gen2 target in the following format:

```
<target>
<target_1>
<target_2>
```

Consider the following rules and guidelines for target partitioning:

- When you read from a directory with multiple partitions and configure target partitioning, the partition files are written to the target based on the number of partitions in the source. However, if you change the partitions in the source and run the mapping task again, ensure that you verify the existing partition files to avoid inconsistent data in the target.
- When you read data with multiple partitions and configure target partitioning, ensure that the target file name is unique and does not match the _part file name from any of the previous mapping runs. Otherwise, the target file might contain inconsistent data.
- You can use the append write strategy only for flat files.
- When you read from a parquet file and write to partitions in a Microsoft Azure Data Lake Storage Gen2 target, the recommended heap size for each partition is 0.5 GB.

File formatting options

Select the format of the Microsoft Azure Data Lake Storage Gen2 file and configure the formatting options.

The following table describes the formatting options for Avro, Parquet, JSON, ORC, and delimited flat files:

Property	Description
Schema Source	The schema of the source or target file. Select one of the following options to specify a schema: - Read from data file. Imports the schema from a file in Microsoft Azure Data Lake Storage Gen2. - Import from schema file. Imports the schema from a schema definition file in the agent machine.
Schema File	The schema definition file in the agent machine from where you want to upload the schema. You cannot upload a schema file when you create a target at runtime.

The following table describes the formatting options for flat files:

Property	Description
Flat File Type	The type of flat file. Select one of the following options: - Delimited. Reads a flat file that contains column delimiters. - Fixed Width. Reads a flat file with fields that have a fixed length. You must select the file format in the **Fixed Width File Format** option. If you do not have a fixed-width file format, click **New** > **Components** > **Fixed Width File Format** to create one.
Delimiter	Character used to separate columns of data in a delimited flat file. You can set values as comma, tab, colon, semicolon, or others. You cannot set a tab as a delimiter directly in the **Delimiter** field. To set a tab as a delimiter, you must type the tab character in any text editor. Then, copy and paste the tab character in the **Delimiter** field.

Property	Description
EscapeChar	Character immediately preceding a column delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string data in a delimited flat file. When you write data to Microsoft Azure Data Lake Storage Gen2 and specify a qualifier, by default, the qualifier is considered as the escape character. Else, the character specified as the escape character is considered.
Qualifier	Quote character that defines the boundaries of data in a delimited flat file. You can set qualifier as single quote or double quote.
Qualifier Mode	Specify the qualifier behavior when you write data to a delimited flat file. You can select one of the following options: - Minimal. Default mode. Applies qualifier to data enclosed within a delimiter value or a special character. - All. Applies qualifier to all data. - Non_Numeric. Not applicable. - All_Non_Null. Not applicable.
Disable escape character when a qualifier is set	Applicable to a Microsoft Azure Data Lake Storage Gen2 target. Select to disable the escape character when a qualifier is set. When you disable the escape character, the special characters not escaped and are considered as part of the data written to the target.
Code Page	Select the code page that the Secure Agent must use to read or write data to a delimited flat file. Select UTF-8 for mappings. Select one of the following options for mappings in advanced mode: - UTF-8 - MS Windows Latin 1 - Shift-JIS - ISO 8859-15 Latin 9 (Western European) - ISO 8859-3 Southeast European - ISO 8859-5 Cyrillic - ISO 8859-9 Latin 5 (Turkish) - IBM EBCDIC International Latin-1
Header Line Number	Specify the line number that you want to use as the header when you read data from a delimited flat file. Specify the value as 0 or 1. To read data from a file with no header, specify the value as 0.
First Data Row[1]	Specify the line number from where you want the Secure Agent to read data in a delimited flat file. You must enter a value that is greater or equal to one. To read data from the header, the value of the **Header Line Number** and the **First Data Row** fields should be the same. Default is 1.
Target Header	Select whether you want to write data to a target that contains a header or without a header in the delimited flat file. You can select **With Header** or **Without Header** options. This property is not applicable when you read data from a Microsoft Azure Data Lake Storage Gen2 source.
Distribution Column	Not applicable.
Max Rows To Preview	Not applicable.

Property	Description
Row Delimiter	Character used to separate rows of data. You can set values as \r, \n, and \r\n. This property is not applicable when you read data from a Microsoft Azure Data Lake Storage Gen2 source.
[1]Doesn't apply to mappings in advanced mode.	

The following table describes the formatting options for JSON files:

Property	Description
Data elements to sample[1]	Specify the number of rows to read to find the best match to populate the metadata.
Memory available to process data[1]	The memory that the parser uses to read the JSON sample schema and process it. The default value is 2 MB. If the file size is more than 2 MB, you might encounter an error. Set the value to the file size that you want to read.
Read multiple-line JSON files	Not applicable.
[1]Applies only to mappings in advanced mode.	

Fixed-width file formats

You can use a fixed-width flat file as a source or target in mappings and mapping tasks.

When you configure a Source transformation or Target transformation and select the fixed-width flat file type, you must select the most appropriate fixed-width file format to use based on the data in the fixed-width flat file. Ensure that the sample flat file only uses UTF-8 character set encoding.

Consider the following rules and guidelines for a fixed-width flat file:

- You cannot use a fixed-width flat file as a source or target for mappings in advanced mode and data transfer tasks.
- When you select a fixed-width file format in the target transformation to create a flat file target at runtime, the Secure Agent ignores the fixed-width column boundaries for the target and considers the fixed-width column boundaries specified for the source.
- When you write a column of Numeric data type from fixed-width flat file source to an empty fixed-width flat file target that uses the same fixed-width file format as the source file, the Secure Agent appends a null character to the value in Numeric column in the target.
- When you create the fixed-width file format, ensure that the sample file uses the following character as the new line symbol, based on the operating system where the Secure Agent is installed:

  - For Linux, use \n character.

  - For Windows, use \r\n character.

  The source file must also use the same character as defined in the sample file.
- When you use a fixed-width flat file as a source or target, you cannot edit the metadata for the fields.

- When you read data of the date data type, you can read the date only up to milliseconds.
- When you read data of the double data type from a fixed-width file and write the data to a Parquet or Avro file, the double data type is mapped to the decimal data type in target. Hence, the data is written incorrectly.
  To write the data correctly to the target, edit the metadata in the Target transformation and change the decimal data type to double.
- When you read from a fixed-width file with a header and write to a fixed-width file and map the FileName field, the header is considered as part of the data. To skip the header, set the value of the **Number of rows to skip** field to 1 when you create the fixed-width file format.
- When you write data to a fixed-width file, you cannot append data to an existing file.

FileName field

A FileName field is a string field that contains the source path of a file. The default precision for a FileName field is 255 characters for a flat file and 1024 characters for a complex file.

You cannot configure the FileName field. You can delete the FileName field if you do not want to read or write the data in the FileName field. You cannot create a folder name with more than 255 characters for a flat file and 1024 characters for a complex file.

FileName is a reserved keyword. Avoid using FileName as the column name in the source data. The name is case sensitive.

The FileName field is applicable to the following file formats:

- Flat file
- Avro
- Parquet
- ORC

Reading source objects path

When you import source objects, the Secure Agent appends a FileName field to the imported source object. The FileName field stores the absolute path of the source file from which the Secure Agent reads the data at run time.

For example, a directory contains a number of files and each file contains multiple records that you want to read. You select the directory as source type in the Microsoft Azure Data Lake Storage Gen2 source advanced properties. When you run the mapping, the Secure Agent reads each record and stores the absolute path of the respective source file in the FileName field.

When you use the FileName field in a source object, the Secure Agent reads file names differently for mappings in advanced mode.

Format type	Syntax	Example
Complex file	`abfss://<filesystem_name>@<account name>.<endpoint>/<directory>/<source_file_name>`	`abfss://adapterqa-source@adlsgen2qa.dfs.core.windows.net/parquet/reader/customer.parquet`
Flat file	`<directory>/<source_file_name>`	`csv/customer.csv`

Writing to target objects

When you import target objects, the Secure Agent appends a FileName field to the imported target object. When you map the FileName field in the target object to an incoming field, the Secure Agent creates the folder structure and the target files based on the FileName field.

The following table describes how the Secure Agent reads file names for mappings:

Format type	Syntax	Example
Complex file	`<directory>/<target_file_name>/<target_file_name>=<values of filename field>/part_file`	`parquet/writer/customer.parquet/customer.parquet=1`
Flat file	`<directory>/<target_file_name>/<target_file_name>=<values of filename field>/part_file` This syntax is applicable only to mappings. If you create a mapping in advanced mode, the Secure Agent does not create a directory structure.	`csv/customer.csv/customer.csv=1`

The following table describes the syntax and example for the FileName field scenarios in mappings:

Description	Syntax	Example
When there is no target, a new target file is created.	`<target_file_name>=<source_file_name>`	`customer_tgt.csv=customer_src.csv`
When the FileName field of an source is mapped to the FileName field of an existing target, a new target file is created and the existing target is not affected.	`<target_file_name>=<source_field_value>`	`customer_tgt.csv=customer_src.csv`
When a source field other than the FileName field is mapped to the FileName field of an existing target, separate files are created for each unique value of the source field.	`<target_file_name>=<source_field_value>`	`customer_tgt.csv=a`, `customer_tgt.csv=b...` `customer_tgt.csv=n` In the example, `n` number of files are inserted into the directory where the target file is present. Where, `n` equals the number of unique values of the source field.

Rules and guidelines for FileName field

Consider the following guidelines when you use the FileName field in mappings:

- FileName is a reserved keyword. Avoid using FileName as the column name in the source data. The name is case sensitive.
- Do not map the source object FileName field to the target object FileName field for a complex file. If you map the FileName field in the target object to an incoming field, the Secure Agent does not create directory structure as expected.
- When you use the FileName field in a target object, the Secure Agent creates folders with different names for null values for mappings in advanced mode:

  - For mappings, the target file name is appended with `_EMPTY_`.

  - For mappings in advanced mode, the target file name is appended with `_HIVE_DEFAULT_PARTITION_`
- When you map a date type incoming field to the FileName field in the target object, the Secure Agent creates a nested folder structure based on the incoming date value for target objects.
- When you map an incoming field to the FileName field in the target object and the target has the **Handle Special Characters** option enabled or the target file name has special characters, the mapping fails with the following error on a Windows machine:
  `[ERROR] java.io.IOException: The filename, directory name, or volume label syntax is incorrect`

- When you map an incoming field to the FileName field in the target object, the mapping runs successfully for the first time. At subsequent runs, the mapping fails with the following error:
  `Object not found.`

  To successfully rerun the mapping, use a dummy target file at design time and override the dummy target file in advanced target properties.
- When you create a target at runtime, the target file name is not generated in the expected format. This issue occurs if the FileName field is enabled for the target object. To resolve this issue, exclude the FileName field from the incoming fields in target.
- When you write data to an existing target object or create a new target object with a partition directory, the FileName field is not added on the target side. The FileName field is only present at the source side. To read the FileName field data from the source, use the Expression transformation to rename the FileName field to a different name to avoid any validation failures.

Directory-level partitioning

You can read from and write to partition columns when you create a mapping in advanced mode.

You can organize tables or data sets into partitions for grouping same type of data together based on a column or partition key. You can select one or more partition columns in a table or data set.

To read from partition columns, select a partition directory and identify the partition columns. To write to partition columns, you can add partition columns from the list of fields and change the partition order, if required.

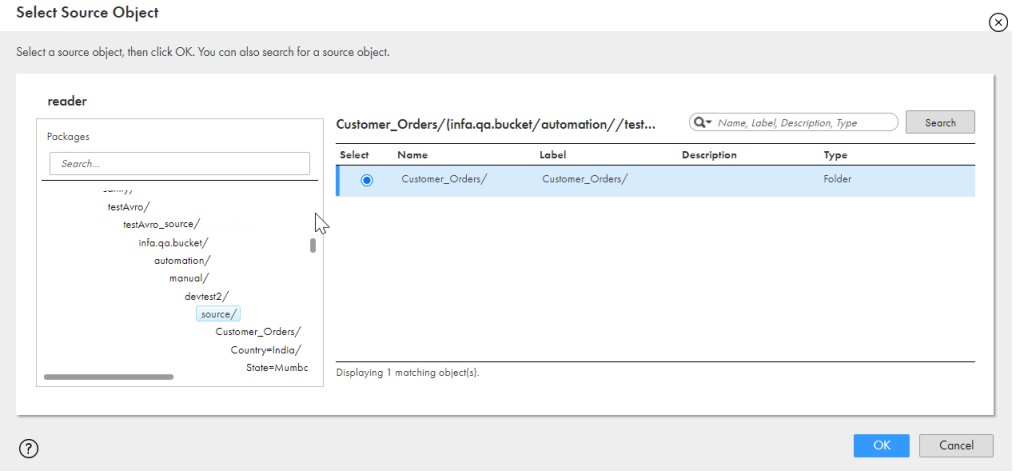
You can read data from or write data to partition columns for the following file formats:

- Avro
- Parquet
- ORC
- JSON

Reading from partition columns

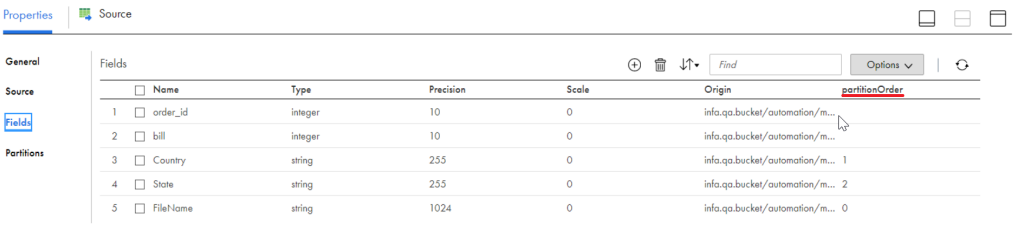
Perform the following steps to read data from partition columns:

1. Select a directory from the list of source objects.



2. Select the Source Type as **Directory** in the Advanced Source Properties.

3. In the Fields tab, you can view the number of partitions.
The **partitionOrder** column indicates whether a column is partitioned and the order in which the fields are selected for partitioning.

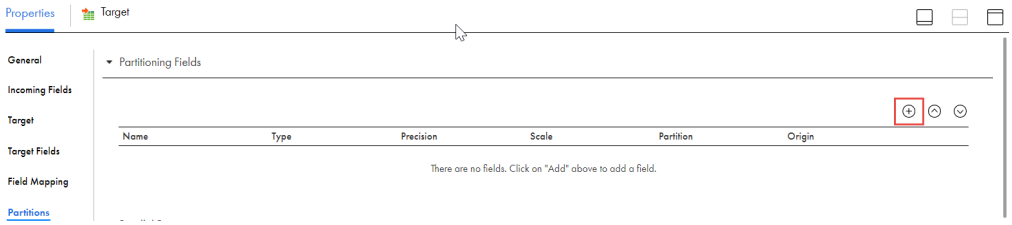


In the above image, 2 partition columns are present. The partition order values 1 and 2 signify the order in which the `Country` and `State` fields were selected for partitioning. The FileName field has 0 as the partition order.

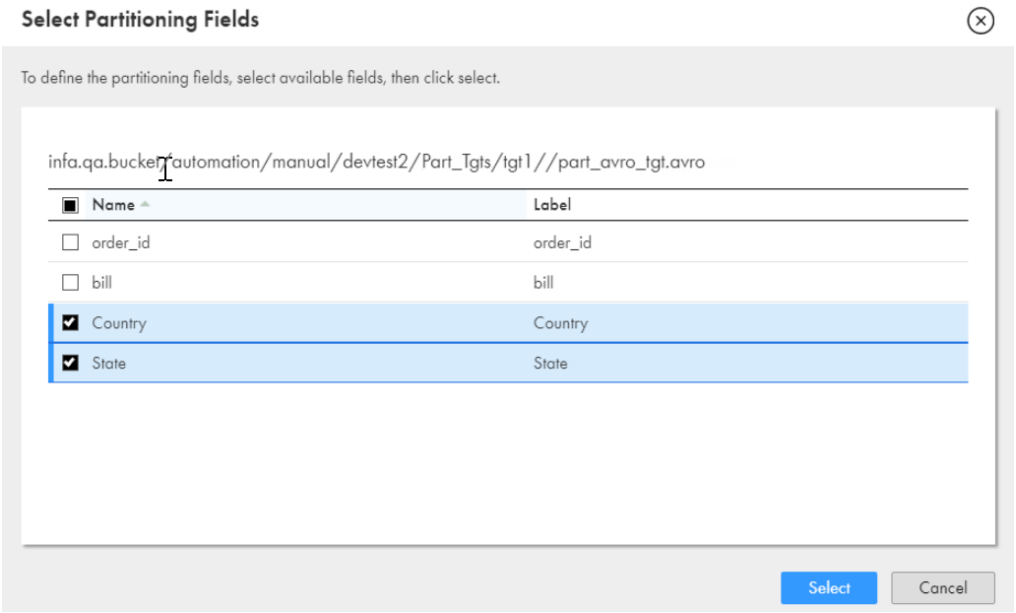
Writing to partition columns

Perform the following steps to write to partition columns:

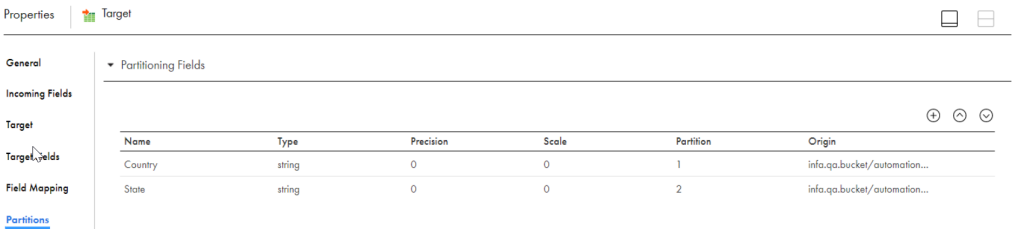
1. In the **Partitions** tab, click **Add** to add the partition columns for a target.



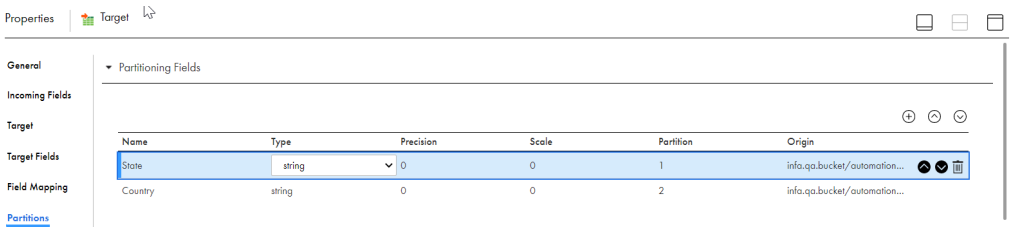
2. Select the partitioning fields from the list of available fields.



3. Click **Select**.
   The Partitions tab shows the partition columns that you select.



You can change the partition order using the up and down arrows as shown in the following image:



Rules and guidelines for reading from and writing to a partition folder

Consider the following rules and guidelines when you read from and write to a partition folder for mappings in advanced mode:

- You must import a directory that contains only partition folders and select the source type as **Directory** in the advanced source property.
- If you import a partition directory that does not have data, a validation error is encountered.
- If you import a partition directory that has a partition folder but no files in the partition folder, a validation error is encountered.
- The FileName field has 0 as the partition order.
- The partitioned directory that you select cannot have a partitioned column named FileName. The name is case insensitive.
- When you import an existing target object or create a new target object with a partition directory, the FileName field is not added on the target side. The FileName field is only present at the source side.
- When you pass a timestamp value in a partition column, the value gets encoded. For example, 03:26:01 is encoded as 03%3A26%3A01.
- When you pass a special character in a partition column, the value gets encoded. For example, #" is encoded as %23%22
- When you import a directory that has a partition folder, the data type for the partition column is imported as a String.
- You cannot edit the data type for a partition column.
- You cannot use the **Edit Metadata** option with partition columns.
- You cannot use the **View Schema** option for a partition directory at source and target side.
- You cannot use the **Import from Schema File** option for partition directory at source because the schema file does not have information for partition columns.
- You cannot use the **Data Preview** option with partition columns.
- You cannot select the partition columns in a mapping task if the target object is parameterized.
- For **Create Target**, you can add partition fields and arrange the partition columns in an order.
- When you import a Microsoft Azure Data Lake Storage Gen2 object that has partition columns, the partition fields are listed at the end of the list.
- If a partition column contains data that has more than 255 characters, the data is truncated and only 255 characters are written in the partition column.
- If a partition column name contains more than 74 characters, the name is truncated and only 74 characters are written in the partition column name.
- The value of the partition directory file path formed using the combination of the partition column name and the target file within the partition directory must not exceed 1024 characters. Otherwise, the mapping will fail.
- You cannot use the **File Name Override** option with partition columns.
- When you read or write JSON files, you cannot use the **Data elements to sample** and **Memory available to process data** advanced properties with partition columns.

Parameterization

You can parameterize the connection, objects, and the advanced runtime properties in mappings.

To parameterize the connection, objects, and the advanced runtime properties using a parameter file, create the parameters in the Parameters panel when you create a mapping. Then, define the parameters in the parameter file, place the parameter file in the following location, and run the mapping task:

```
<Informatica Cloud Secure Agent\apps\Data_Integration_Server\data\userparameters>
```

You can also save the parameter file in a cloud-hosted directory in Microsoft Azure Data Lake Storage Gen2.

You cannot save the parameter file in a cloud-hosted directory for a mapping in advanced mode.

Consider the following rules and guidelines when you use parameterization:

General guidelines

- When you create a mapping with a parameterized target that you want to create at runtime, set the target field mapping to automatic.
- You cannot parameterize the field mapping.

Mappings

- When you use input parameters, specify the parameter name in the following format:
  - Format in a mapping task: $name$
  - Format in a parameter file: $name
- When you use in-out parameters, specify the parameter name in the following format in a mapping task or a parameter file: $$name.
- You cannot parameterize a Microsoft Azure Data Lake Storage Gen2 target created at runtime. Instead, you can specify the parameter in the Directory Override to parameterize the target using a parameter file. Specify the parameter in the following format: $$name or $name.

Mappings in advanced mode

- When you use input parameters, specify the parameter name in the following format in a mapping task: $name$.
- When you use input parameters in a parameter file to parameterize the connection or object, the parameter value is stored in the following format: $$name.
- You cannot use input parameters in a parameter file to parameterize the advanced properties at runtime.
- Ensure that you create the parameter before you override it using a parameter file.
- You cannot parameterize only the connection. You can either paramaterize both the connection and the object or only the object.
- You cannot parameterize hierarchical data types.
- When you parameterize the object, ensure that the metadata of the imported object matches the metadata fetched in the override.
- When you use the parameter file to parameterize the source or target objects, you must specify the absolute path of the objects.
  For example, `$$SrcObj=<Directory1>/.../<DirectoryN>/<FileName>;`
- When you parameterize the connection or object, ensure that the connection or object that you want to override exists in Microsoft Azure Data Lake Storage Gen2.

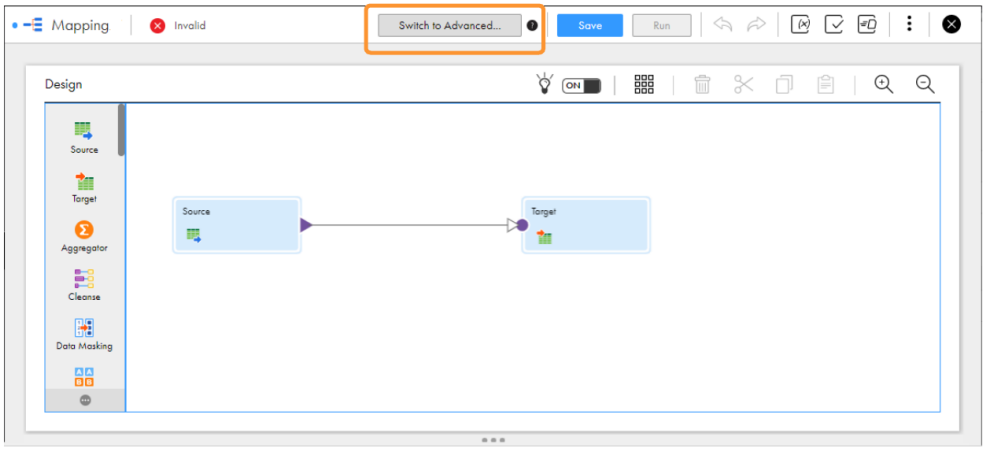
Mappings in advanced mode example

You work for one of the largest community college that maintains millions of records in their ongoing student database. The college has more than 10,000 faculty members teaching at 45 campuses and 700 locations across the globe. The college has a very large IT infrastructure and about 15 TB of information gets downloaded on daily basis from the Internet.

To avoid performance, scalability, and high cost challenges, the college plans to port its entire data from its operational data stores to Microsoft Azure Data Lake Storage Gen2 within a short span of time. Configure a mapping in advanced mode to achieve faster performance when you read data from the operational data stores and write data to the Microsoft Azure Data Lake Storage Gen2 target.

1. In Data Integration, click **New** > **Mappings** > **Mapping**.
2. In the Mapping Designer, click **Switch to Advanced**.

   The following image shows the **Switch to Advanced** button in the Mapping Designer:



3. In the **Switch to Advanced** dialog box, click **Switch to Advanced**.

   The Mapping Designer updates the mapping canvas to display the transformations and functions that are available in advanced mode.
4. Enter a name, location, and description for the mapping.
5. On the Source transformation, specify a name and description in the general properties.
6. On the **Source** tab, perform the following steps to provide the source details to read data from the source:
   a. In the **Connection** field, select the required source connection.
   b. In the **Source Type** field, select the type of the source.
   c. In the **Object** field, select the required object.
   d. In the **Advanced Properties** section, provide the appropriate values.

   Note: When you read a flat file, ensure that the file has a header row. Else, the Secure Agent fails to read the first row.
7. On the **Fields** tab, map the source fields to the target fields.
8. On the Target transformation, specify a name and description in the general properties.

9. On the **Target** tab, perform the following steps to provide the target details to write data to the Microsoft Azure Data Lake Storage Gen2 target:
    a. In the **Connection** field, select the Microsoft Azure Data Lake Storage Gen2 target connection.
    b. In the **Target Type** field, select the type of the target.
    c. In the **Object** field, select the required object.
    d. In the **Operation** field, select the required operation.
    e. In the **Advanced Properties** section, provide appropriate values for the advanced target properties.
10. Map the source and target.
11. Click **Save** > **Run** to validate the mapping.

    In Monitor, you can monitor the status of the logs after you run the task.

Rules and guidelines for mappings

Consider the following rules and guidelines for mappings:

Mappings

- When you configure a mapping, ensure that you use only the special characters that are allowed in Microsoft Azure Data Lake Storage Gen2 in the directory name, file name, or object path.
  For more information about the special characters allowed in Microsoft Azure Data Lake Storage Gen2, see the Microsoft Azure documentation.
- You cannot read and write primitive data types in nested and multi-line indented JSON files in mappings.
- When a column name in the source starts with a number and you create a target at runtime, the corresponding target column is prefixed with an underscore character (_).
- When you create a Microsoft Azure Data Lake Storage Gen2 target at runtime to write an Avro, ORC, or Parquet file, you cannot write null values with primitive data types.
- When you use the JVM options to configure the HTTPS proxy server and read and write Avro, Parquet, and ORC files, the mapping fails with the following error:
  `AzureADAuthenticator.getTokenCall threw java.io.IOException : Unable to tunnel through proxy. Proxy returns "HTTP/1.1 407 Proxy Authentication Required"`
- When you append data to a Microsoft Azure Data Lake Storage Gen2 target, you must map all the incoming fields to target fields.
- When you create a Microsoft Azure Data Lake Storage Gen2 target at runtime and specify the path in the object name field, ensure that you specify the complete path including the file system name.
  For example, `<FileSystem_Name>/<Directory1>/.../<DirectoryN>/<FileName>`
- When the data has more than one escape character and you do not select the **Disable escape char when a qualifier is set** option, all the escape characters are not written to the flat file target in Microsoft Azure Data Lake Storage Gen2.
  For example, `"Ga\\lit",124 "Ga\\\"l",19` is written as `Ga\lit,124 "Ga\$"l",19`.
- You cannot read hierarchical data types in a mapping. You can use the Hierarchy Parser transformation to convert the hierarchical input into relational output.
  For more information about configuring a Hierarchy Parser transformation, see *Transformations* in the Data Integration documentation.

- When you run a mapping to write a JSON file to a Microsoft Azure Data Lake Storage Gen2 target, the mapping writes the values of double data type in exponential format in the target.
- When you run a mapping configured with a fixed partition of 8 to read from a parquet file of size 20 GB or more and write to a Microsoft Azure Data Lake Storage Gen2 target, the mapping fails with the following error:
  `[ERROR] java.lang.OutOfMemoryError: Java heap space`
- If a mapping includes parameterized source and target, **Allow parameter to be overridden at run time** checkbox is selected, and the source object selected resides in a folder during the mapping task creation, the mapping fails with the following error:
  `[ERROR] Exception: Exception occured in read phase, error: Exception while downloading file to local staging`
- When you use the Snappy compression format to write data to Microsoft Azure Data Lake Storage Gen2, the mapping retains a **snappy-1.1.8****-libsnappyjava.so** file in the temp directory on the agent machine after it runs successfully.
- When you create a new target and select **Handle Special Characters** to append time stamp to the file name, the file name override is not honored in a mapping.
- When you run a mapping to create a Microsoft Azure Data lake storage Gen2 target, use the append strategy, and enable **Handle Special Characters**, the file is either created or appended based on the timestamp you include in the file name.
- When you parameterize the target connection and object and create a new target at runtime, the target is created in the root directory specified in the Connection Directory path even if you specify the complete path in the create target object name.
  To resolve this, you can specify an absolute or relative path in Directory Override in advanced target properties to create the target in the overriden path.
- When you read and write complex files, set the JVM options for type DTM to increase the -Xms and -Xmx values in the system configuration details of the Secure Agent to avoid java heap space error. The recommended -Xms and -Xmx values are 512 MB and 1024 MB respectively.
- When you read and write complex files, ensure that the file name does not contain percentage (%) or hash (#). Otherwise, the data preview and mapping fails at runtime.

Mappings in advanced mode

- When you run a mapping in advanced mode to read data from a Microsoft Azure Data lake storage Gen2 source, use a parameter file to parameterize the source connection and object, and specify the directory and file override in the advanced properties, the mapping considers the values specified in the parameter file.
- When you read data from and write data to Microsoft Azure Data Lake Storage Gen2, use the same storage account for both the source and target connections.
  If you want to use different storage accounts, use shared authentication for one account and service principal authentication for the other account. You cannot use the same authentication type for both the storage accounts.
- When you read data from and write data to Microsoft Azure Data Lake Storage Gen2 and use the shared key authentication, ensure that you use the same access key for both the source and target connections.
  For example, if you use Key 1 as the access key for the source connection and Key 2 for the target connection, the mapping fails.
- When you use the managed identity authentication, you cannot use system assigned identity.

- When you use managed identity authentication, ensure that the storage account specified in the connection is not the same as the storage account specified in the staging location and log location for the Azure cluster.
- You can read and write hierarchical data types for Avro, JSON, and Parquet files. You can also read hierarchical data types for ORC files.
- When you set the qualifier mode to Minimal and use an escape character, the escape characters are not escaped and quoted in the target. To resolve this issue, set the qualifier mode to All.
- When you set the qualifier mode to All and do not specify a value for the qualifier, \00 (Null) is considered as the qualifier.
- You cannot add multiple pipelines in a mapping.
- When you read from a complex file source of size 128 MB or more, the Secure Agent writes incorrect data and creates multiple target files without overriding the existing target.
- You cannot read zero-byte files when you run mappings in advanced mode.
- When you upload a schema file for the source and create a Microsoft Azure Data Lake Storage Gen2 target at runtime, ensure that source file is not empty.
- When you append data to an existing target, you must configure any overrides in the advanced target properties, else the mapping fails.
- When you append data to a target created at runtime and if a file with the same name exists in the target directory, the mapping fails.
  In this case, you must first overwrite the existing file and then append the data.
- When a JSON file has a field with empty struct data, the Secure Agent ignores the field and reads the remaining fields during metadata read.
  For example, if the JSON file has the following data in the first row: `{"id":123,"address":{}}`, the `address` field is ignored and does not appear in the **Fields** tab. If the JSON file has values for the `address` field in the consecutive row, you can use the **Data elements to sample** property to fetch this field.
- When you run a mapping in advanced mode and map the source fields of double or float data type to the target fields of string data type, the format of the values changes in the target.
  The following table describes the change in the format of the values in the target:

Value in the source	Value in the target
1.7976931348623157e+308	1.79769313486232e+308
-9999999999999.99	-10000000000000
4.94065645841247e-324	4.9e-324
7956318123.99392483	7956318123.99392

CHAPTER 4

Migrating a mapping

You can configure a connection and mapping in one environment and then migrate and run the mapping in another environment.

You can also migrate mappings configured in advanced mode. After the migration, you can change the connection properties from the Administrator service, but you do not need to modify the mapping. Data Integration uses the configured runtime attributes from the earlier environment to run the mapping successfully in the new environment.

Consider a scenario where you develop a mapping in the development organization (Org 1) and you then migrate and run the mapping in the production organization (Org 2). After you migrate, you might want to use the same or a different connection endpoint or object path in Org 2. Based on your requirement, follow the guidelines in this section before you plan the migration.

Use the same object path for the migrated mapping

If you want the migrated mapping in Org 2 to use the same object path as in Org 1, you must maintain the same file system, directory, and file name in the Microsoft Azure Data Lake Storage Gen2 account for Org 2.

For example, if you have two different accounts, Account1 used for Org 1 and Account2 used for Org 2, the object path for the file system, directory, and file name must be the same in both the accounts:

Account1: FileSystem1/Directory1/FileName1

Account2: FileSystem1/Directory1/FileName1

In this scenario, you do not need to override the file system, directory, and file name in the advanced properties and the mapping runs successfully.

Use a different object path for the migrated mapping

After you migrate the mapping, you can use a different object path to run the mapping from the new environment.

In this scenario, before you migrate the mapping, you can change the object metadata, runtime attributes, or the connection attributes to reflect the object path in the migrated environment. You do not have to edit or update the mapping in the new environment.

As a rule, when you specify the file system, directory, and file name in the advanced properties, connection, or object properties, Data Integration honors the attributes in the following order of precedence:

1. **Runtime advanced attributes**. The advanced properties such as file system, directory, and file name in the Source or Target transformation in a mapping.
2. **Connection attributes**. Attributes such as file system and directory set in the connection properties.
3. **Object metadata**. The object selected in the Source or Target transformation in a mapping.

Migration options

When you migrate, you can choose from one of the following options to update the object path:

Option 1. Update the connection properties to reference the new object

When you import the mapping into Org 2, in the **Review Connections** section, you can change the existing connection to map to the connection that has access to the specified file system, directory, and file name in Org 2.

Option 2. Override the properties from the advanced properties

Before the migration, specify the required file system, directory, and file name for the object from Org 2 in the advanced properties of the Org 1 mapping.

After the migration, when you run the mapping, the Secure Agent uses the configured advanced parameters to override the object specified in the mapping imported from Org 1.

Option 3. Parameterize the advanced properties

You can choose to parameterize the advanced attributes, such as the file system, directory, and file name before the migration. You can configure input parameters, in-out parameters, and parameter files in the mapping. After you migrate the mapping, do not edit or update the mapping. If you have used in-out parameters for the advanced attributes such as for the file system, directory, and file name, you can update these from the parameter file.

Parameterizing only the advanced properties, but not the object in the mapping

If you want to parameterize only the advanced properties and use them at runtime, select a placeholder object in the object properties in the mapping and then specify an override to this placeholder object from the advanced properties. Ensure that the placeholder object contains the same metadata as the corresponding table that you specify as an override. When you run the mapping, the value specified in the advanced property overrides the placeholder object.

Parameterizing both the object and the advanced properties

If you want to keep both the Microsoft Azure Data Lake Storage Gen2 object type and the advanced fields parameterized, you must leave the **Allow parameter to be overridden at runtime** option unselected in the input parameter window while adding the parameters, and then select the required object at the task level. When you run the task, the values specified in the advanced properties take precedence.

Parameterization rules

Consider the following rules to parameterize the object and advanced properties:

- Parameterization is not applicable for mappings that use the **Create Target** option.
- Parameterization is not applicable for mappings in advanced mode in the migration use case.
- If the parameter file is saved in a cloud-hosted directory in Microsoft Azure Data Lake Storage Gen2, after the migration, do not change the directory path or the parameter file name.

- If there are multiple pipelines configured in a mapping, do not parameterize the Microsoft Azure Data Lake Storage Gen2 object. You must select a placeholder object while creating the mapping before you migrate.

General rules and guidelines

Consider the following rules and guidelines when you migrate a mapping:

- When you migrate a mapping, ensure that the metadata of the imported object matches the metadata fetched in an override.
- When you migrate a mapping, do not specify the relative directory in the directory override field. When you migrate and run the mapping, the path used for importing the object during the design time is added to the specified file path while fetching the metadata at runtime.
- When you configure a mapping to create a target at runtime and enable the dynamic schema handling option, and then upon migration, you change the source connection in the mapping for which the FileName field is deleted in the database, the FileName field is still written to the target even though it was deleted.
  You must exclude the FileName field from the Target transformation to resolve this issue.
- When you use a different object path for the migrated mapping, Data Integration uses the imported object metadata instead of the metadata fetched at runtime in the following cases:
  - Gzip compression is enabled in advance source properties.
  - FileName field is mapped.
- When you enable the Recursive Directory Read option and the first level directory contains multiple folders but no files, after you migrate and run the mapping, Data Integration uses the imported object metadata instead of the metadata fetched at runtime.

CHAPTER 5

Data type reference

Data Integration uses the following data types in Microsoft Azure Data Lake Storage Gen2 mappings and mapping tasks:

- Microsoft Azure Data Lake Storage Gen2 native data types appear in the Source transformation and Target transformation when you choose to edit metadata for the fields.
- Transformation data types. Set of data types that appear in the transformations. These are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. They appear in all transformations in a mapping.

When the Secure Agent reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the Secure Agent writes to a target, it converts the transformation data types to the comparable native data types.

The following table lists the Microsoft Azure Data Lake Storage Gen2 data types that Data Integration supports and the corresponding transformation data types:

Microsoft Azure Data Lake Storage Gen2 Native Data Type	Transformation Data Type	Description
String	String	1 to 104,857,600 characters

Flat file data types and transformation data types

Flat file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the flat file data types that the Secure Agent supports and the corresponding transformation data types:

Flat file data type	Transformation data type for mappings	Transformation data type for mappings in advanced mode	Range and description
BigInt	Not applicable	Long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 characters; precision 19, scale 0
Nstring*	Text	Text	1 to 104,857,600 characters
Number*	Decimal	Double	Precision from 1 through 28 digits, scale from 0 through 28 digits
String*	String	String	1 to 104,857,600 characters. Precision 256.
*You must select the **Schema Source** as **Import from schema file** to read data of Number, String, or Nstring data type.			

Avro data types and transformation data types

Avro file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the Avro file data types that the Secure Agent supports and the corresponding transformation data types:

Avro Data Type	Transformation Data Type	Range and Description
Array[1]	Array	Unlimited number of characters
Boolean	Integer	1 or 0 True is equivalent to the integer 1 and False is equivalent to the integer 0.
Bytes	Binary	Precision 4000
Double	Double	Precision 15
Float	Double	Precision 15
Int	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0

Avro Data Type	Transformation Data Type	Range and Description
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
Map[1]	Map	Unlimited number of characters
Null	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Record[1]	Struct	Unlimited number of characters
String	String	1 to 104,857,600 characters Precision 4000
Union[1]	Corresponding data type in a union of ["primitive_type|complex_type", "null"] or ["null", "primitive_type|complex_type"].	Dependent on primitive or complex data type.
[1]Applies only to mappings in advanced mode.		

JSON data types and transformation data types

JSON file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the JSON file data types that the Secure Agent supports and the corresponding transformation data types:

JSON Data Type	Transformation Data Type	Range and Description
Array[1]	Array	Unlimited number of characters
boolean	integer	The default transformation type for boolean is integer. You can specify string data type with values of True and False. True is equivalent to the integer 1 and False is equivalent to the integer 0.
Number (double)	double	-1.79769313486231570E+308 to +1.79769313486231570E+308. Precision 15.
Number (float)	double	-1.79769313486231570E+308 to +1.79769313486231570E+308. Precision 15.
Number (int)	integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Number (long)	bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0.

JSON Data Type	Transformation Data Type	Range and Description
Object[1]	Struct	Unlimited number of characters
string	string	1 to 104,857,600 characters. Precision 4000
[1]Applies only to mappings in advanced mode.		

ORC data types and transformation data types

ORC file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the ORC file data types that the Secure Agent supports and the corresponding transformation data types:

ORC File Data Type	Transformation Data Type	Range and Description
BigInt	BigInt	-9223372036854775808 to 9,223,372,036,854,775,807
Boolean	Integer	1 or 0 True is equivalent to the integer 1 and False is equivalent to the integer 0.
Char	String	1 to 104,857,600 characters
Date	Date/Time	Jan 1, 1753 A.D. to Dec 31, 4712 A.D. (precision to microsecond)
Double	Double	Precision of 15 digits
Float	Double	Precision of 15 digits
Integer	Integer	-2,147,483,648 to 2,147,483,647
SmallInt	Integer	-32,768 to 32,767
String	String	1 to 104,857,600 characters Precision 4000
Timestamp	Date/Time	1 to 19 characters Precision 19 to 26, scale 0 to 6

ORC File Data Type	Transformation Data Type	Range and Description
TinyInt	Integer	-128 to 127
Varchar	String	1 to 104,857,600 characters

Parquet data types and transformation data types

Parquet file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the Parquet file data types that the Secure Agent supports and the corresponding transformation data types:

Parquet data type	Transformation data type	Range and description
Boolean	Integer	1 or 0 True is equivalent to the integer 1 and False is equivalent to the integer 0.
Byte_Array	Binary	Arbitrarily long byte array
Date	Date/Time	January 1, 0001 to December 31, 9999.
Decimal	Decimal	Precision 1 to 28 digits, scale 0 to 28. You cannot use decimal values with precision greater than 28.
Double	Double	Precision 15
Float	Double	Precision 15
group(LIST)[1]	Array	Unlimited number of characters.
Int32	Integer	-2,147,483,648 to +2,147,483,647
Int64	Bigint	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 8-byte signed integer
Int96	Binary	12-byte signed integer When you configure a mapping in advanced mode, the data preview displays the time zone of the Secure Agent machine.

Parquet data type	Transformation data type	Range and description
Map[1]	Map	Unlimited number of characters.
String	String	1 to 104,857,600 characters Precision 4000
Struct[1]	Struct	Unlimited number of characters.
Time	Date/Time	Time of the day. Precision to microsecond.
Timestamp	Date/Time	January 1, 0001 00:00:00 to December 31, 9999 23:59:59.997. Precision to microsecond. You cannot set the precision to nanoseconds.
[1]Applies only to mappings in advanced mode.		

The Parquet schema that you specify for the Parquet file must be in smaller case. Parquet does not support case-sensitive schema.

Parquet timestamp data type support

You can use the following Timestamp data types for Parquet file format:

Timestamp Data type	Mappings	Mappings in advanced mode
Timestamp_micros	Yes	No
Timestamp_millis	Yes	No
Time_millis	Yes	No
Time_micros	Yes	No
int96	Yes	Yes

You cannot use the following Timestamp data types for Parquet file format:

- Timestamp_nanos
- Time_nanos
- Timestamp_tz

CHAPTER 6

Troubleshooting

Use the following sections to troubleshoot errors in mappings.

Troubleshooting a mapping

Time zone for the Date and Timestamp data type fields in Parquet or Avro file formats defaults to the Secure Agent host machine time zone.

When you run a mapping to read from or write to fields of the Date and Timestamp data types in the Parquet or Avro file formats, the time zone defaults to the Secure Agent host machine time zone.

To change the Date and Timestamp to the UTC time zone, configure the JVMOptions in the Secure Agent.

Perform the following steps to configure the JVM options in the Secure Agent:

1. Select **Administrator** > **Runtime Environments**.
2. On the **Runtime Environments** page, select the Secure Agent for which you want to configure the JVMOptions.
3. In the upper-right corner, click **Edit**.
4. In the **System Configuration Details** section, select **Data Integration Server** as the service and **DTM** as the type.
5. Edit the **JVMOption** field and set the value to `-Duser.timezone=UTC`.
6. Click **Save**.

Mapping failed with a Java heap space error

When you read from or write to large data sets in Microsoft Azure Data Lake Storage Gen2, certain mappings might fail with the following error:

```
[ERROR] java.lang.OutOfMemoryError: Java heap space
```

You must increase the heap size to run the mappings successfully. The recommended heap size is 1 GB.

Perform the following steps to configure the JVM options in the Secure Agent to increase the memory for the Java heap size:

1. Select **Administrator** > **Runtime Environments**.
2. On the **Runtime Environments** page, select the Secure Agent for which you want to increase memory from the list of available Secure Agents.
3. In the upper-right corner, click **Edit**.

4. In the **System Configuration Details** section, select **Data Integration Server** as the service and **DTM** as the type.
5. Edit the **JVMOption** field and set the value to **-Xmx1024m**.

   Note: The recommended heap size is 1 GB. You can increase the heap size based on the data you want to process.
6. Click **Save**.

Unicode character strings are written incorrectly to the target

When you run a Microsoft Azure Data Lake Storage Gen2 mapping on a Linux machine, parameterize the source connection and object, and append data to the existing target, the Unicode character strings in the data are not written correctly to the target.

To resolve this issue, set the environment variable `LC_ALL="en_US.UTF-8"` in the Secure Agent on the Linux machine, and restart the Secure Agent.

Mapping fails if the directory name or subdirectory name contains Unicode characters

When you read from or write to a flat file in Microsoft Azure Data Lake Storage Gen2 and if the directory name or subdirectory name contains Unicode characters, the mapping fails.

To resolve this issue, set the environment variable `LC_ALL="en_US.UTF-8"` in the Secure Agent, and restart the Secure Agent.

Non-English characters in the source are incorrectly written to the target when you use the Append write strategy

When you append data to a flat file in a Microsoft Azure Data Lake Storage Gen2 target, the non-English characters in the source are incorrectly written to the target.

To write the non-English characters correctly to the target, configure the JVMOptions in the Secure Agent.

Perform the following steps to configure the JVM options in the Secure Agent:

1. Select **Administrator** > **Runtime Environments**.
2. On the **Runtime Environments** page, select the Secure Agent for which you want to configure the JVMOptions.
3. In the upper-right corner, click **Edit**.
4. In the **System Configuration Details** section, select **Data Integration Server** as the service and **DTM** as the type.
5. Edit the **JVMOption** field and set the value to `-Dfile.encoding=UTF-8`.
6. Click **Save**.

Mapping fails due to failure to access the /tmp directory

When you run a Microsoft Azure Data Lake Storage Gen2 mapping, the Secure Agent stages the files in a temporary staging folder. By default, the folder for staging data is `/tmp`.

Ensure that you have the read and write permissions to the `/tmp` folder.

Perform the following steps to change the temporary staging folder:

1. Select **Administrator** > **Runtime Environments**.
2. On the **Runtime Environments** page, select the Secure Agent for which you want to configure the JRE_OPTS field.
3. In the upper-right corner, click **Edit**.
4. In the **System Configuration Details** section, select **Data Integration Server** as the service and **DTM** as the type.

5. Edit the **JRE_OPTS** field and set the value to `-Djava.io.tmpdir=<DIR>`.
6. Click **Save**.

Troubleshooting a mapping in advanced mode

Mapping configured to read or write Date and Int96 data types for Avro or Parquet files fails

A mapping configured to read from or write to an Avro or Parquet file fails in the following cases:

- Data is of the Date data type and the date is less than 1582-10-15.
- Data is of the Int96 data type and the timestamp is less than 1900-01-01T00:00:00Z.

To resolve this issue, specify the following spark session properties in the mapping task or in the custom properties file for the Secure Agent:

- `spark.sql.legacy.timeParserPolicy=LEGACY`
- `spark.sql.parquet.int96RebaseModeInWrite=LEGACY`
- `spark.sql.parquet.datetimeRebaseModeInWrite=LEGACY`
- `spark.sql.parquet.int96RebaseModeInRead=LEGACY`
- `spark.sql.parquet.datetimeRebaseModeInRead=LEGACY`
- `spark.sql.avro.datetimeRebaseModeInWrite=LEGACY`
- `spark.sql.avro.datetimeRebaseModeInRead=LEGACY`

Time zone for the Date and Timestamp data type fields in Parquet or Avro file formats defaults to the Secure Agent host machine time zone.

When you run a mapping to read from or write to fields of the Date and Timestamp data types in the Parquet or Avro file formats, the time zone defaults to the Secure Agent host machine time zone.

To change the Date and Timestamp to the UTC time zone, you can either set the Spark properties globally in the Secure Agent directory for all the tasks in the organization that use this Secure Agent, or you can set the Spark session properties for a specific task from the task properties:

To set the properties globally, perform the following tasks:

1. Add the following properties to the `<Secure Agent installation directory>/apps/At_Scale_Server/41.0.2.1/spark/custom.properties` directory:
   - infacco.job.spark.driver.extraJavaOptions=-Duser.timezone=UTC
   - infacco.job.spark.executor.extraJavaOptions=-Duser.timezone=UTC
2. Restart the Secure Agent.

To set the properties for a specific task, navigate to the Spark session properties in the task properties, and perform the following steps:

- Select the session property name as `spark.driver.extraJavaOptions` and set the value to `-Duser.timezone=UTC` .
- Select `spark.executor.extraJavaOptions` and set the value to `-Duser.timezone=UTC`.

INDEX

C

Cloud Application Integration community
  URL 5
Cloud Developer community
  URL 5
connections
  Microsoft Azure Data Lake Storage Gen2 9, 10
create target
  adding time stamps 25
  target file parameterization 25

D

Data Integration community
  URL 5
data type reference
  overview 44
data types
  avro 45
  parquet 48
directory source
  Microsoft Azure Blob Storage sources 18

I

incrementally load files
  overview 20
Informatica Global Customer Support
  contact information 6
Informatica Intelligent Cloud Services
  web site 5

J

JSON file data types
  transformation data types 46

M

maintenance outages 6
mapping in advanced mode
  example 37
mappings
  Microsoft Azure Data Lake Storage Gen2 Source properties 15
  Microsoft Azure Data Lake Storage Gen2 target properties 21
Microsoft Azure Data Lake Storage Gen2
  connection properties 10
Microsoft Azure Data Lake Storage Gen2 *(continued)*
  Source transformation 15
  Sources in mappings 15
  Target transformation 21
  targets in mappings 21
Microsoft Azure Data Lake Storage Gen2 Connection
  overview 9
Microsoft Azure Data Lake Storage Gen2 connector
  example 7
Microsoft Azure Data Lake Storage Gen2 Connector
  introduction 7

O

ORC file data types
  transformation data types 47

S

Source transformation
  Microsoft Azure Data Lake Storage Gen2 properties 15
Sources
  Microsoft Azure Data Lake Storage Gen2 in mappings 15
specifying targets 24
status
  Informatica Intelligent Cloud Services 6
system status 6

T

Target transformation
  Microsoft Azure Data Lake Storage Gen2 properties 21
targets
  Microsoft Azure Data Lake Storage Gen2 in mappings 21
trust site
  description 6

U

upgrade notifications 6

W

web site 5
wildcard character
  overview 19