Informatica® Cloud Data Integration

Microsoft CDM Folders V2 Connector

Informatica Cloud Data Integration Microsoft CDM Folders V2 Connector
August 2023

Table of Contents

Preface

Use *Microsoft CDM Folders V2 Connector* to learn how to read from or write to CDM folders on Microsoft Azure Data Lake Storage Gen2. Learn to create a connection, develop and run mappings, mapping tasks, and data transfer tasks in 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, go to https://status.informatica.com/ and click **SUBSCRIBE TO UPDATES**. You can then 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 Customer Support Center by telephone or online.

For online support, click **Submit Support Request** in Informatica Intelligent Cloud Services. You can also use Online Support to log a case. Online Support requires a login. You can request a login at https://network.informatica.com/welcome.

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.

Introduction to Microsoft CDM Folders V2 Connector

You can use Microsoft CDM Folders V2 Connector to read data from or write data to CDM folders on Microsoft Azure Data Lake Storage Gen2.

You can create a common data model folder within a file system in the ADLS Gen2 storage. A common data model folder contains data in the `.csv` file format and the associated metadata in the `model.json` and `manifest.json` file formats. You can read from CDM folders that contain metadata in the `model.json` or `manifest.json` format and write to CDM folders that contain metadata in the `manifest.json` format.

You can use Microsoft CDM Folders V2 Connector to read from and write to CDM format files. When Microsoft CDM Folders V2 Connector writes to CDM format files, you can use a CDM Aware application that uses the latest CDM SDK to access the CDM format files.

You can use Microsoft CDM Folders V2 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.

For more information about the common data model, see the Microsoft documentation at the following website:

https://docs.microsoft.com/en-us/common-data-model/

Microsoft CDM Folders V2 Connector assets

Create assets in Data Integration to integrate data using Microsoft CDM Folders V2 Connector.

When you use Microsoft CDM Folders V2 Connector, you can include the following Data Integration assets:

- Data transfer task
- Mapping
- Mapping task

Note: You can only use the insert operation when you write data to a Microsoft CDM Folders V2 target.

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

Standard common data model and custom objects

You can use CDM Folders Connector to store the data as standard common data model objects or custom objects.

The standard common data model objects are predefined objects in the common data model such as Account. Applications such as Microsoft Dynamics use the common data model objects for integration across their business applications and processes.

If the data that you want to store does not match with any of the standard common data model objects, you can create new objects to store the data. You can create new objects with custom names and fields from the source when you create a target using the **Create Target** option.

Administration of Microsoft CDM Folders V2 Connector

Before you use Microsoft CDM Folders V2 objects in tasks, an administrator must perform the following tasks:

- Create an ADLS Gen2 account and provide **Contributor** or **Reader** role to users.
  - Using the contributor role, you can have full access to manage all resources in the storage account, but you cannot assign roles.
  - Using the reader role, you can view all resources in the storage account, but you cannot make any changes.

  To add or remove role assignments, you must have write and delete permissions, such as an Owner role.
- Create an Azure Active Directory application to authenticate users to access the ADLS Gen2 account. Provide **Storage Blob Data Contributor** or **Storage Blob Data Reader** role to the application.
  - Using the Storage Blob Data Contributor role, you can read, write, and delete Azure Storage containers and blobs in the storage account.
  - Using the Storage Blob Data Reader role, you can only read and list Azure Storage containers and blobs in the storage account.

  To write to or delete Azure Storage containers and blobs, you must have the Contributor role either at the storage account level or the container level.
- Create a file system for ADLS Gen2.
- Create an Azure AD web application for service-to-service authentication with ADLS Gen2.

  Note: Ensure that you have superuser privileges to access the folders or files created in the application using the connector.

Connections for Microsoft CDM Folders V2

Create a Microsoft CDM Folders V2 connection to securely read data from or write data to CDM folders on Microsoft Azure Data Lake Storage Gen2. You can use a Microsoft CDM Folders V2 connection to specify sources and targets in mappings and mapping tasks.

Microsoft CDM Folders V2 connection properties

When you set up a Microsoft CDM Folders V2 connection, configure the connection properties.

The following table describes the Microsoft CDM Folders V2 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	The Microsoft CDM Folders V2 connection type.
Runtime Environment	The name of the runtime environment where you want to run the tasks. Select a Secure Agent, Hosted Agent, or serverless runtime environment.
ADLSGen2 Storage Account Name	Name of the ADLS Gen2 storage account.
Azure AD App Client ID	The client ID of the Azure Active Directory account to authenticate user access to the storage account. You can get the application ID from the Microsoft Azure Active Directory administrator.
Azure AD App Client Secret	The client secret key of the Azure Active Directory application to authenticate access to the storage account. You can get the key value from the Microsoft Azure Active Directory administrator.

Property	Description
Azure Tenant ID	The tenant ID of the Azure Active Directory account to authenticate user access to the storage account. You can get the directory ID from the Microsoft Azure Active Directory administrator.
ADLSGen2 File System Name	The name of the file system that you create in the Azure Storage Explorer application. A file system can contain more than one common data model folders.
CDM Folder Path	The path of the common data model folder that you create within the file system. You can use the following values for CDM folder path: - `/` - `/folder1` - `/folder1/folder2` The recommended CDM folder path is `/folder1`. Default is empty.
Adls Gen2 End-point	The ADLS Gen2 endpoint core.windows.net.

CHAPTER 3

Mappings for Microsoft CDM Folders V2

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 CDM Folders V2 object. Use the Mapping Designer in Data Integration to add the Source or Target transformations in the mapping canvas and configure the Microsoft CDM Folders V2 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 CDM Folders V2 sources in mappings

In a mapping, you can configure a source transformation to represent a single Microsoft CDM Folders V2 object.

The following table describes the Microsoft CDM Folders V2 source properties that you can configure in a source transformation:

Property	Description
Connection	Name of the source connection.
Source Type	Type of the CDM folder source objects. Select **Single Object** or **Parameter**.
Object	Name of the source object. Select an existing object.

The following table describes the Microsoft CDM Folders V2 source advanced properties that you can configure in a Source transformation:

Property	Description
Concurrent Threads	Number of concurrent connections to read data from CDM Folders V2. When you read from a large file or object, consider spawning multiple threads to process data. Default is 10.
Block Size	Divides a large file or object into smaller parts each of specified block size. When you read from a large file, consider dividing the file into smaller parts and configure concurrent connections to spawn the required number of threads to process data in parallel. Default is 8 MB.
File System Name Override	Overrides the default file system name at run time with the file system name that you specify.
CDM Folder Override	Overrides the default CDM folders. You can use the following paths for CDM folder override: - `/` - `/folder1` - `/folder1/folder2` - `/folder1/filename`, for example, `/folder1/default.manifest.cdm.json` or `/folder1/model.json`
Entity Name Override	Overrides the default entity name in the `manifest.json` or `model.json` file.
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.

Note: You cannot use special characters in an entity name.

Overriding the source advanced properties

You can choose to override the default file system name, CDM folder, or entity name in the source advanced properties.

During mapping development you might use connection attributes, and during run time you can change the attributes without changing the connection attributes.

The CDM folder path and entity name must be valid when you override the default file system name, CDM folder, or entity name. Otherwise, the mapping fails with an error.

You can override the file system name when you select the file system name override option. For example, if you configured connection attributes for file system name `cdmsd`, CDM folder path `demo/model`, and the entity name `ff_cdm`, you can change the attributes during run time. You can select the file system name override and specify the system name `cdmsd_prod`. However, `cdmsd_prod` must have the same CDM folder path and the entity name that you specified in Microsoft CDM Folders V2 connection.

You can also override the filename when you select the CDM folder override option. For example, if you specified the CDM folder path `/folder1/default.manifest.cdm.json` in the Microsoft CDM Folders V2 connection to read from `manifest.cdm.json` file, you can choose to read from `model.json` file during run time.

Microsoft CDM Folders V2 source transformation in mappings example

In the following example, when you configure a Source transformation, you can select the Microsoft CDM Folders V2 connection to connect to the ADLS Gen2 storage and read data from a source object on the common data model folder.

1. Create a CDM Folders V2 source connection to read data from the common data model folder in the ADLS Gen2 storage.
2. Create a CDM Folders V2 mapping.
3. Add a Source transformation to include the source object and perform the following tasks on the **Source** tab:
   a. In the **Connection** field, select the CDM Folders V2 connection to connect to the common data model folder in the ADLS Gen2 storage.
   b. Select a folder that contains the `default.manifest.cdm.json` or `model.json` file. You can read from the `default.manifest.cdm.json` or `model.json` file.
   c. Select the `default.manifest.cdm.json` or `model.json` file that contains the object you want to read from.
   d. Select an existing source object.
      The following image shows the existing objects in the `default.manifest.cdm.json` file:
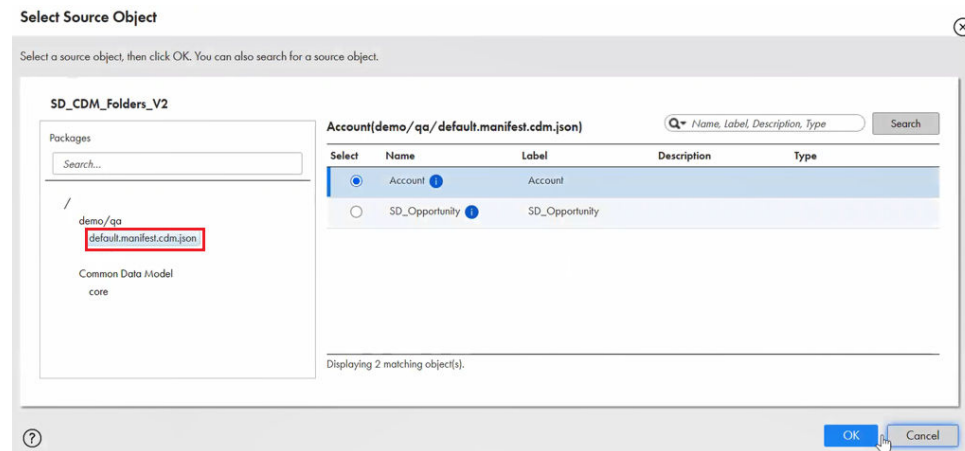
   e. In the **Advanced Properties** section, set the tracing level to normal.
4. Add a Target transformation to include the target object.

Microsoft CDM Folders V2 target transformation in mappings

When you configure a target, you can choose to write to a standard object or a custom object.

When you write to a standard object, you can select an existing target object. When you write to a custom object, you can select an existing target object or create a new object as a target object at runtime.

You can write to files that contain metadata in the `manifest.json` format.

Writing to standard objects

The `Common Data Model` object folder contains all the supported standard objects from the common data model. All the standard objects from the common data model are placed in a hierarchical folder structure defined by Microsoft.
Select the target object as **Existing** and then select objects from the `Common Data Model` object folder if you want to write data to standard common data model objects.

The following image shows the `Common Data Model` object folder that displays all the supported standard objects from the common data model:
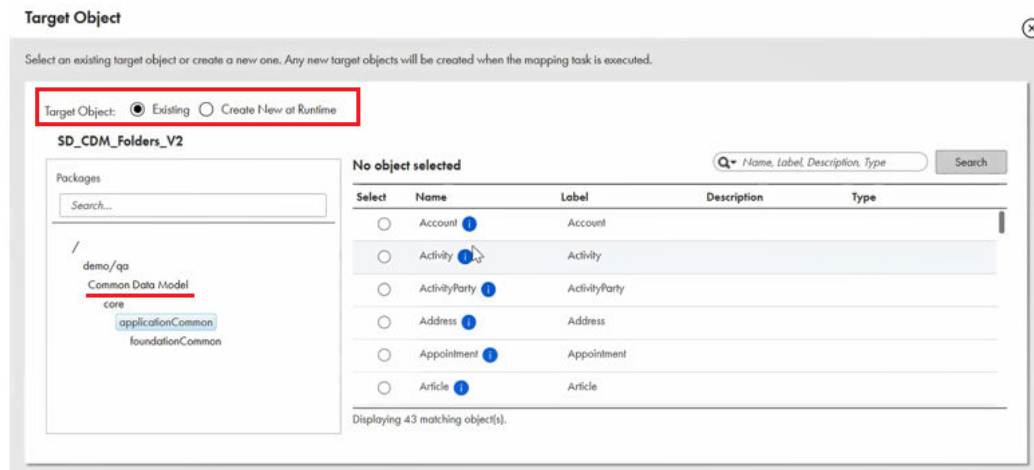
You can use standard objects available in the `Common Data Model` object folder when the CDM folder that you specified in the connection properties, for example, `demo/qa` does not contain any predefined objects.

When you run the mapping, the following files are added to the CDM folder, for example, `demo/qa`:

- The Account folder contains the actual data.
- `Account.cdm.json` contains all the metadata information such as fields and data types of the fields.
- `default.manifest.cdm.json` file contains information about all the objects.
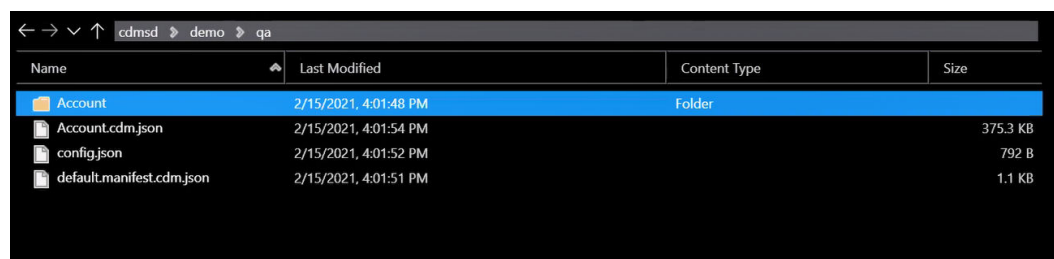
The following image shows the manifest file that contains information about the Account object:

```
                                                              default.manifest.cdm.json
{
    "jsonSchemaSemanticVersion" : "1.1.0",
    "imports" : [
        {
            "corpusPath" : "cdm:/foundations.cdm.json"
        }
    ],
    "manifestName" : "tempAbstract",
    "entities" : [
        {
            "type" : "LocalEntity",
            "entityName" : "Account",
            "entityPath" : "Account.cdm.json/Account",
            "dataPartitionPatterns" : [
                {
                    "name" : "DataPartitionPattern",
                    "rootLocation" : "Account",
                    "globPattern" : "/*Account*.csv",
                    "exhibitsTraits" : [
                        {
                            "traitReference" : "is.partition.format.CSV",
                            "arguments" : [
                                {
                                    "name" : "columnHeaders",
                                    "value" : "true"
                                },
                                {
                                    "name" : "delimiter",
                                    "value" : "|"
                                },
                                {
                                    "name" : "escape",
                                    "value" : "\""
                                },
                                {
                                    "name" : "quote",
                                    "value" : "\""
```

Write strategy for standard objects

When you select the **Overwrite** write strategy and run the mapping, the object folder, for example, `Account`, and `<objectname>_cdm.json` file, for example, the `Account.cdm.json` file are replaced or created.

When you select the **Append** write strategy and run the mapping, the Secure Agent appends the data to the object in ADLS Gen2 with a new object name in the following `.csv` file format:

`objectname_{unique_ID}.csv`.

The following image shows the object created in CDM Folders when you run a mapping using the append write strategy:
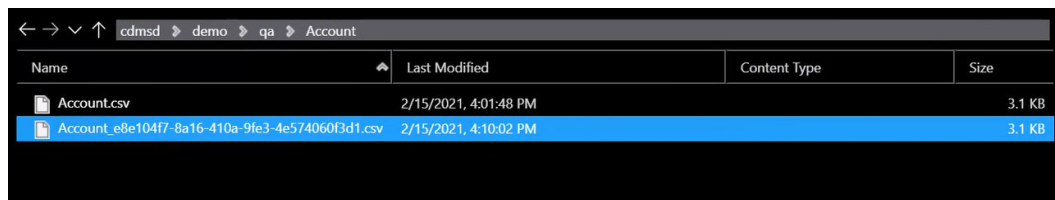
Writing to custom objects

You can select an existing object from the `default.manifest.cdm.json` file. You can also create a new object a runtime when the source object is not available in the list of the standard common data model objects or when the schema of the source object does not match the schema of the standard object.

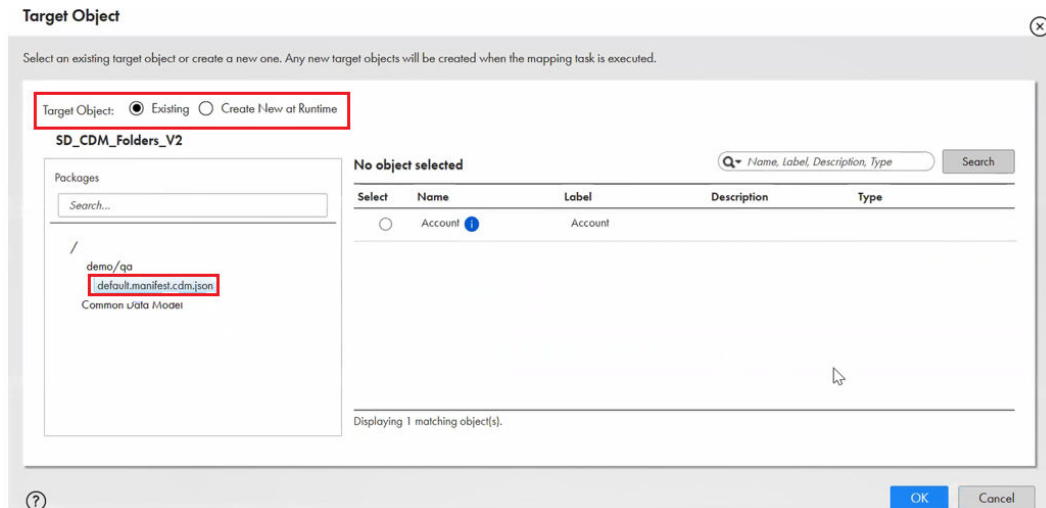**Selecting an existing target object**

Select the target object as **Existing** to select an existing object.

Select the CDM folder path that you specified in the connection properties. For example, `demo/qa`.

The CDM folder path shows the `default.manifest.cdm.json` file that contains all the existing objects.

The following image shows the CDM folder that displays all the existing objects:
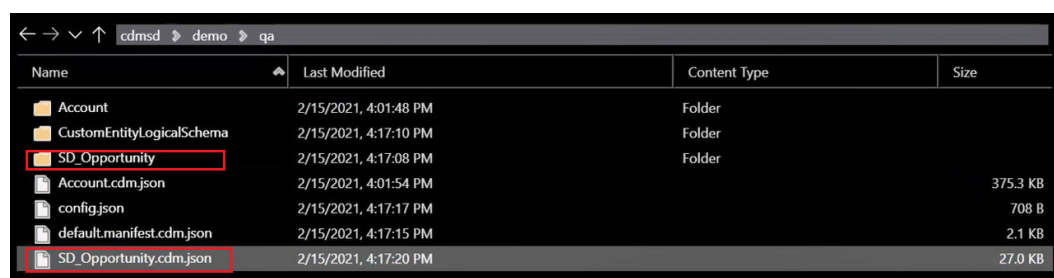
Creating a new target object

Select the target object as **Create New at Runtime** to create a new target object.
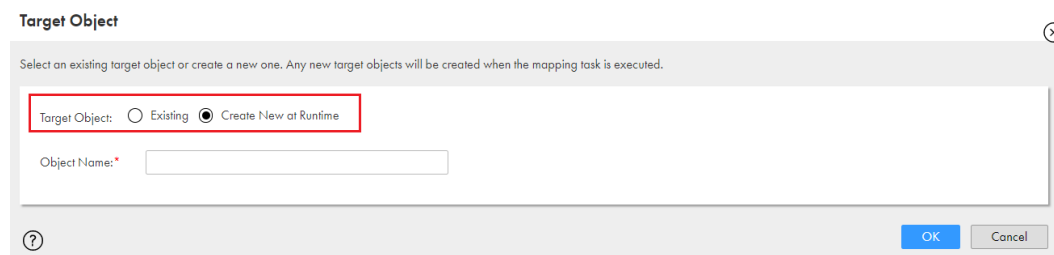
Write strategy for custom objects

When you create a target object, for example, `SD_Opportunity`, and select the **Append** write strategy, the Secure Agent appends the data to the object in ADLS Gen2.

In the image, the following folders are created:

- `SD_Opportunity` contains the data.

- `SD_Opportunity.cdm.json` and `Account.cdm.json` are the resolved schema documents for the objects.

- `CustomEntityLogicalSchema` is the directory created for custom objects and contains the logical schema documents.

When you select the **Overwrite** write strategy and select the **CDM folder Override** option in the target properties, a subdirectory, for example, `override` is created.
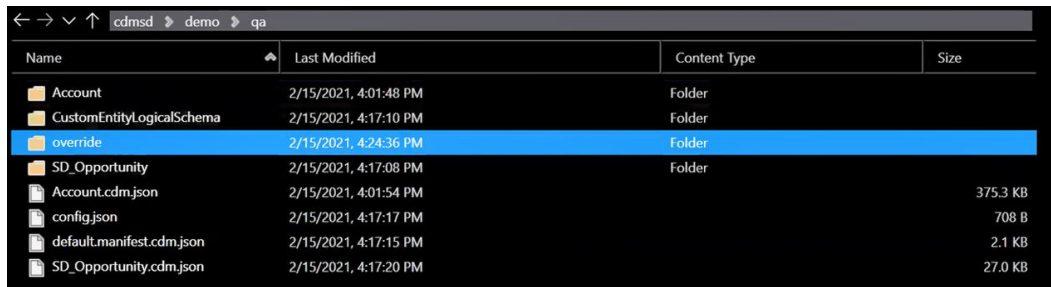
A folder path is created if it does not exist and the entity name is replaced with the entity that you create.
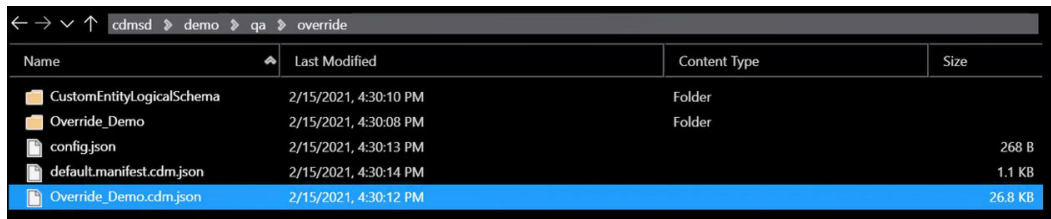
The following image shows the entity `Override_Demo` that overrides the entity `SD_Opportunity`:

Note: Do not edit the `default.manifest.cdm.json` and `cdm.json` files to prevent data inconsistencies.

Microsoft CDM Folders V2 targets in mappings

In a mapping, you can configure a Target transformation to represent a Microsoft CDM Folders V2 target. You can configure the target and advanced target properties in the **Targets** page of the Mapping Task wizard.

The following table describes the Microsoft CDM Folders V2 target properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the CDM Folders V2 target connection.
Target Type	Type of the CDM Folders target objects. You can select **Single Object** or **Parameter**.
Object	Name of the target object. Select an existing object or create a new object.
Operation	Type of the target operation. You must select the **Insert** operation.

The following table describes the Microsoft CDM Folders V2 target advanced properties that you can configure in a Target transformation:

Property	Description
Concurrent Threads	Number of concurrent connections to write data to CDM Folders V2. When you write to a large file or object, consider spawning multiple threads to process data. Default is 10.
Block Size	Divides a large file or object into smaller parts each of specified block size. When you write to a large file, consider dividing the file into smaller parts and configure concurrent connections to spawn required number of threads to process data in parallel. Default is 8 MB.
File System Name Override	Overrides the default file system name at run time with the file system name you specify.
CDM Folder Override	Overrides the CDM folders during run time. For mapping development, you might use connection attributes, and during run time, change the attributes without changing the connection attributes.
Write Strategy	Determines how you write to the target object. Select **Overwrite** or **Append**. When you select overwrite, the Secure Agent creates a new target object or replaces the existing target object. When you select append, the source object is appended at the target.
FF Parser Properties	Overrides the default flat file parser values during run time. You can choose a column header, delimiter, and a quote character. Specify the values in the following format: `columnHeaders=true` `delimiter=,` `quote="` The flat file parser properties are case sensitive. By default, `delimiter=,`, `quote="`, and the data files are written without headers.
Entity Name Override	Overrides the entity name during the run time.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. By default, the mapping task application forwards rejected rows to the next transformation.

Note: When you create an entity with the default flat file parser properties and select the append operation in a Target transformation, you cannot override the flat file parser properties.

Rules and guidelines for mappings and mapping tasks

Consider the following rule and guideline for mappings and mapping tasks:

- When you create a target at run time and the source file does not contain any row, the task fails with a null pointer exception.

CHAPTER 4

Data type reference

Data Integration uses the following data types in Microsoft CDM Folders V2 mappings and mapping tasks:

- Microsoft CDM Folders V2 native data types appear in the Source transformationand 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.

Microsoft CDM Folders V2 and transformation data types

The following table lists the supporting Microsoft CDM Folders V2 data types and the corresponding transformation data types:

Microsoft CDM Folders V2 Data Type	Transformation Data Type	Description
Boolean	String	TRUE (1) or FALSE (0)
Date	Date	Date values. Microsoft CDM Folders V2 Connector uses the following format: yyyy-MM-dd Maximum precision 29, scale 9. Default precision 29.
DateTime	Date/Time	Date values. Microsoft CDM Folders V2 Connector uses the following format: yyyy-MM-dd'T'HH:mm:ssZ Maximum precision 29, scale 9. Default precision 29.
Decimal	Decimal	Maximum precision 31, scale 0. Default precision 31.

Microsoft CDM Folders V2 Data Type	Transformation Data Type	Description
Double	Double	Maximum precision 15, scale 0. Default precision 15.
Guid	String	Maximum precision 255, scale 0. Default precision 255.
int64	BigInteger	Maximum precision 20, scale 0. Default precision 20.
int32	Integer	Maximum precision 11, scale 0. Default precision 11.
int16	Integer	Maximum precision 10, scale 0. Default precision 10.
Json	String	Maximum precision 255, scale 0. Default precision 255.
String	String	Maximum precision 255, scale 0. Default precision 255.

CHAPTER 5

Troubleshooting

Use the following section to troubleshoot errors in Microsoft CDM Folders V2 Connector.

SDK exception when entity name has Unicode (UTF-8) characters

When you run a mapping to read from a CDM Folders V2 source object where the entity name contains UTF-8 characters, the task fails with the following error:

```
"com.informatica.adapter.sdkadapter.exceptions.AdapterSDKException: [SDK_APP_COM_20000] error
[No Data Partitions found indefault.manifest.cdm.json]"
```

To resolve this issue, perform the following tasks to set `-Dfile.encoding=UTF8` in the JVM option in the Secure Agent:

1. Select **Administer** > **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 the **Type** as **DTM** for the Data Integration Service.
5. Edit the **JVMOption** as **-Dfile.encoding=UTF8**.
6. Restart the Secure Agent manually.

Network latency error when you write a large file

When you run a mapping task to write a large file, you might encounter the following error:

```
"ERROR Operators: Operator called default onErrorDroppedjava.lang.IllegalStateException"
```

This issue occurs due to network latency.

To resolve this issue, perform the following tasks to increase the `tunnelouttimeperiod`, `NetworkTimeoutPeriod`, and `NetworkRetryInterval`:

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 the **Type** as **Tomcat** for the Data Integration Server.

5. Increase the value of **TunnelTimeoutPeriod**, **NetworkTimeoutPeriod**, and **NetworkRetryInterval**.

   Note: Specify the values based on the data you want to process.

6. Restart the Secure Agent manually.

Overhead limit exceeded error when you write large records

When you run a mapping to write large number of records, you might encounter the following error:

```
"ERROR java.lang.OutOfMemoryError: GC overhead limit exceeded"
```

To resolve this issue, perform the following tasks to configure the JVM options in the Secure Agent to increase the memory for the Java heap size:

1. Select **Administer** > **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 the **Type** as **DTM** for the Data Integration Service.

5. Edit the **JVMOption1** as **-Xms512m** and **JVMOption2** as **-Xmx1024m**.

   Note: Specify the maximum and minimum heap size based on the data you want to process.

6. Restart the Secure Agent manually.

INDEX