



Informatica® Metadata Command Center
November 2025

Microsoft Azure Data Factory Sources

© Copyright Informatica LLC 2024, 2025

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, Informatica Cloud, Informatica Intelligent Cloud Services, PowerCenter, PowerExchange, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2025-12-04

Table of Contents

| | |
|---|-----------|
| Preface | 4 |
| Chapter 1: Introduction to Microsoft Azure Data Factory catalog sources | 5 |
| Extraction and view process | 6 |
| About the Microsoft Azure Data Factory catalog source | 7 |
| Extracted metadata. | 7 |
| Chapter 2: Before you begin | 10 |
| Verify permissions. | 10 |
| Permissions to extract metadata | 10 |
| Permissions to run data classification. | 10 |
| Permissions to run glossary association. | 10 |
| Get Microsoft Azure Data Factory source information. | 10 |
| Chapter 3: Create catalog sources in Metadata Command Center. | 12 |
| Step 1. Register a catalog source. | 12 |
| Step 2. Configure capabilities. | 14 |
| Configure metadata extraction. | 14 |
| Configure lineage discovery. | 22 |
| Configure data classification. | 23 |
| Configure glossary association. | 24 |
| Step 3. Associate stakeholders and asset groups. | 25 |
| Step 4. Run or schedule the job. | 27 |
| Step 5. Assign reference catalog source connections to endpoint catalog source objects. | 28 |
| Chapter 4: View results in Data Governance and Catalog. | 30 |
| View metadata extraction results. | 31 |
| View referenced source systems. | 33 |
| View data lineage. | 33 |
| View lineage at the catalog source level. | 33 |
| View lineage at data set and data element levels. | 34 |
| View classified data. | 36 |
| View glossary associations. | 36 |

Preface

Read *Microsoft Azure Data Factory Sources* to learn how to register and configure Microsoft Azure Data Factory sources in Metadata Command Center as catalog sources. After you configure a catalog source, you extract metadata and then view the results in Data Governance and Catalog.

CHAPTER 1

Introduction to Microsoft Azure Data Factory catalog sources

You can use Metadata Command Center to extract and view metadata from a Microsoft Azure Data Factory source system. A source system is any system that contains data, such as Oracle and Microsoft Azure Data Factory. You can run connection-aware scans on Microsoft Azure Data Factory sources.

Before you extract metadata from a source system, you first create and register a catalog source that represents the source system. Then you configure capabilities for the catalog source. A capability is a task that Metadata Command Center can perform, such as metadata extraction, lineage discovery, data profiling, data classification, or glossary association.

You can also add filters to extract specific sets of metadata. When Metadata Command Center extracts metadata, Data Governance and Catalog displays the extracted metadata and its attributes as technical assets. You can then perform tasks such as analyzing the assets, viewing lineage, and creating links between those assets and their business context.

The Microsoft Azure Data Factory catalog source can use operational metadata captured from activity and pipeline runs to resolve dynamic metadata. In this scenario, lineage is extracted when you use queries in pipelines to extract metadata from database tables. Next, you can use the extracted metadata to concatenate and run dynamic queries.

For example, the lineage is extracted when you use the lookup activity to get a list of tables from a given database and perform one of the following actions:

- Run the 'For Each' activity and use each table value as input for subsequent operations within the pipeline.
- Use the table list as a condition in 'If condition' activities that use the output of the initial lookup.

You can use Databricks Delta Lake as both a source and a target in data flows and pipeline activities.

The following table describes the capabilities of the catalog source:

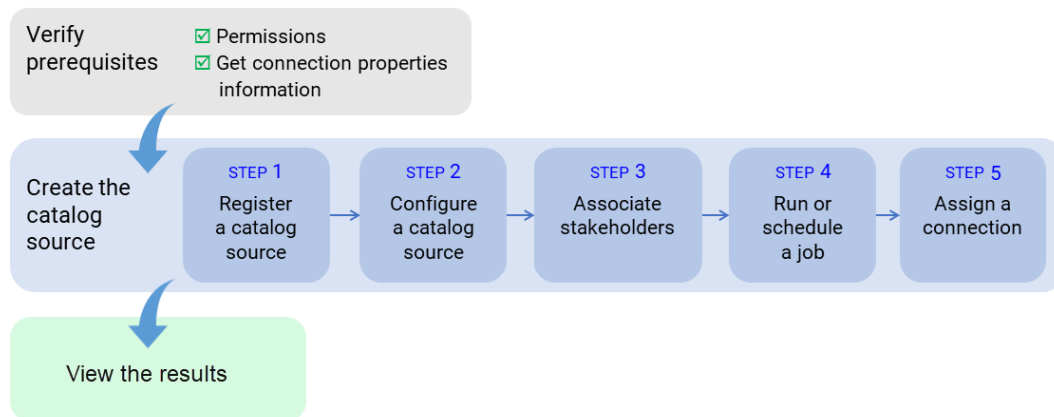
| Capability | Description |
|--------------------------------|---|
| Serverless Runtime Environment | A serverless runtime environment is an advanced serverless deployment solution that doesn't require downloading, installing, configuring, or maintaining a Secure Agent or Secure Agent group. You can use a serverless runtime environment in the same way that you use a Secure Agent when you configure a catalog source. |
| Lineage Discovery | Builds the complete lineage of a catalog source by recommending endpoint catalog source objects to assign to reference catalog source connections. When you run the catalog source job, Metadata Command Center assigns the reference catalog source connections to CLAIRE recommended endpoint catalog source objects. You can then view the list of CLAIRE recommendations and accept or reject them. |

| Capability | Description |
|----------------------|---|
| Data Classification | Data classification is the process of identifying and organizing data into relevant categories based on the functional meaning of the data. Classifying data can help your organization manage risks, compliance, and data security. |
| Glossary Association | You can associate terms that are in the glossary with technical assets to provide user-friendly business names to technical assets. Glossary Association automatically associates glossary terms with technical assets or recommends glossary terms that you can manually associate with technical assets in Data Governance and Catalog. |

Extraction and view process

To extract metadata from a source system, configure the catalog source and run the extraction job in Metadata Command Center. Then view the results in Data Governance and Catalog.

The following image shows the process to extract metadata from a source system:



After you verify prerequisites, perform the following tasks to extract metadata from Microsoft Azure Data Factory:

1. Register a catalog source. Create a catalog source object, select the source system, and specify values for connection properties.
2. Configure the catalog source. Specify the runtime environment and configure parameters for metadata extraction. Optionally, add filters to include or exclude source system assets from metadata extraction. You can also configure other capabilities such as data classification and glossary association.
3. Associate stakeholders. Associate users with technical assets, giving the users permission to perform actions determined by their roles.
4. Run or schedule the catalog source job.
5. Assign a connection to referenced source system assets.

After you run the catalog source job, you view the results in Data Governance and Catalog.

About the Microsoft Azure Data Factory catalog source

You can use the Microsoft Azure Data Factory catalog source to extract metadata from an Microsoft Azure Data Factory source system.

Microsoft Azure Data Factory is a cloud-based ETL and data integration service that is hosted on Microsoft Azure. You can create and configure a Microsoft Azure Data Factory catalog source to extract metadata from the Microsoft Azure Data Factory source system. You can run connection-aware scans on Microsoft Azure Data Factory sources.

Note: To improve wildcard lineage at the directory or file level for Microsoft Azure Data Factory assets, perform connection assignment, and run the Microsoft Azure Data Factory catalog source again. These wildcards can refer to files in Amazon S3 and Microsoft Azure Data Lake Storage Gen2.

Extracted metadata

You can use the Microsoft Azure Data Factory catalog source to extract metadata from a Microsoft Azure Data Factory source system.

Metadata Command Center extracts the following metadata from a Microsoft Azure Data Factory source system:

- Activity
- Calculation
- Pipeline
- Dataset
- Dataflow
- Resource
- Folder
- Resource Group
- Package
- Data Task
- Factory

Pipeline activities and datasets

The Microsoft Azure Data Factory catalog source supports the following Pipeline Activity and Dataset components:

| Component name | Description |
|-----------------|---|
| Copy | Pipeline activity |
| ExecuteDataFlow | Pipeline activity. Partial support. It processes the input and output providing generic lineage. It does not analyze script transformations. |

| Component name | Description |
|--------------------------|--|
| ExecuteSSISPackage | Pipeline activity. Supported page locations: - File System (Package) - Embedded Package |
| GetMetadata | Pipeline activity |
| Lookup | Pipeline activity |
| SqlServerStoredProcedure | Pipeline activity |
| IfCondition | Pipeline activity |
| Switch | Pipeline activity |
| Until | Pipeline activity |
| ForEach | Pipeline activity |
| Wait | Pipeline activity |
| SetVariable | Pipeline activity |
| ExecutePipeline | Pipeline activity |
| WebActivity | Pipeline activity |
| DatabricksNotebook | Pipeline activity |
| Unsupported | Pipeline activity |
| Binary | Dataset |
| DelimitedText | Dataset |
| Excel | Dataset |
| Json | Dataset |
| Parquet | Dataset |
| FileShare | Dataset |
| AmazonRedshiftTable | Dataset |
| AzureSqlDWTable | Dataset |
| AzureSqlTable | Dataset |
| DynamicsCrmEntity | Dataset |
| OracleTable | Dataset |
| SalesforceObject | Dataset |

| Component name | Description |
|---------------------------------|-------------|
| SnowflakeTable | Dataset |
| SqlServerTable | Dataset |
| AzureDatabricksDeltaLakeDataset | Dataset |

CHAPTER 2

Before you begin

Before you create a catalog source, ensure that you have the information required to connect to the source system.

Verify permissions

To extract metadata and to configure other capabilities that a catalog source might include, you need account access and permissions on the source system. The permissions required might vary depending on the capability.

Permissions to extract metadata

To extract metadata from a source system, you need account access and permissions to the Microsoft Azure Data Factory source system.

Verify that the administrator performs the following tasks:

- Creates a user account for the Informatica user to access the Microsoft Azure Data Factory source system.
- Assigns the Informatica user account to the Reader or Monitoring Reader role.
- Grants permissions for the Reader or Monitoring Reader role at the resource group level.

Permissions to run data classification

You can perform data classification with the permissions required to perform metadata extraction.

Permissions to run glossary association

You can perform glossary association with the permissions required to perform metadata extraction.

Get Microsoft Azure Data Factory source information

Before you configure the catalog source, ask the Microsoft Azure Data Factory administrator for property values that you need to configure the catalog source.

The following table describes the properties that you need:

| Property | Description |
|-----------------|---|
| Subscription ID | Required. Subscription ID of Microsoft Azure Data Factory. |
| Client ID | Required. Client or application ID of Microsoft Azure Data Factory. |
| Tenant ID | The URL of the Microsoft Azure Data Factory instances. Required when Microsoft Azure Data Factory uses Service Principal authentication. If the administrator gives you this information, use the Service Principal connection mode when you configure the connection information. |
| Client Secret | The client secret key to connect to Microsoft Azure Data Factory instances. Required when Microsoft Azure Data Factory uses Service Principal authentication. If the administrator gives you this information, use the Service Principal connection mode when you configure the connection information. |
| User name | Fully qualified user name to connect to Microsoft Azure Data Factory instances. Required when Microsoft Azure Data Factory allows user names and passwords for access. If the administrator gives you this information, use the Admin User connection mode when you configure the connection information. |
| Password | Password associated with the user name. Required when Microsoft Azure Data Factory allows user names and passwords for access. If the administrator gives you this information, use the Admin User connection mode when you configure the connection information. |

CHAPTER 3

Create catalog sources in Metadata Command Center

Use Metadata Command Center to configure a catalog source for Microsoft Azure Data Factory and extract metadata.

When you configure a catalog source, you define the source system that you want to extract metadata from. Configure filters to include or exclude source system metadata before you run the job. Optionally, configure other capabilities, such as lineage discovery, data profiling and quality, data classification, relationship discovery, and glossary association.

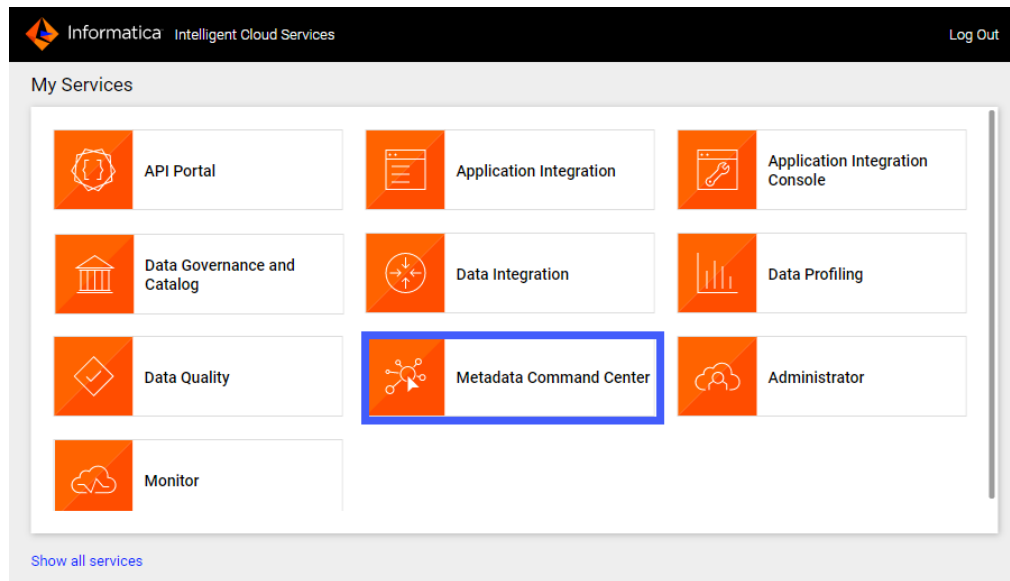
To provide stakeholders access to technical assets, you can assign access through stakeholder roles. You can also associate technical assets extracted from the catalog source to asset groups. If your catalog source references other source systems, you can create a connection assignment to the endpoint catalog source to view complete lineage.

Step 1. Register a catalog source

When you register a catalog source, provide general information and connection values.

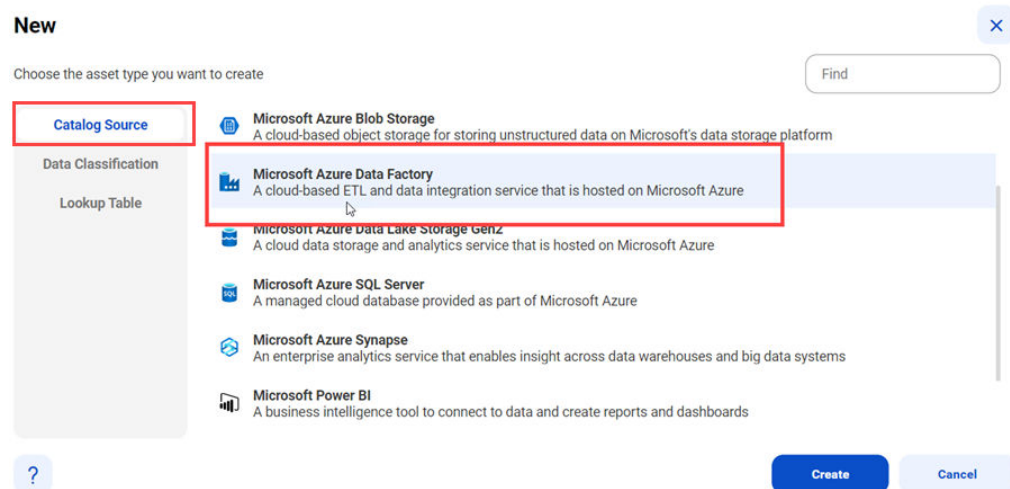
1. Log in to Informatica Intelligent Cloud Services.
The **My Services** page appears.
2. Click **Metadata Command Center**.

The following image shows the Metadata Command Center box on the **My Services** page:



The Metadata Command Center home page appears.

3. Click **New**.
4. Select **Catalog Source** from the list of asset types.
5. Select Microsoft Azure Data Factory from the list of catalog source types and then click **Create**.



The **New Catalog Source** page opens.

6. In the **General Information** section, enter a name and an optional description for the catalog source.

Note: You can rename a catalog source after you create it, but to apply the change to all associated objects you must rerun the metadata extraction job.

After you save the catalog source, you can update the description in Metadata Command Center and Data Governance and Catalog. The update appears only in the service in which you update it.

7. On the **Connection Information** panel, choose the Microsoft Azure Data Factory connection mode based on the connection values that you got from the Azure administrator.

Choose to connect as the Admin User or by using Service Principal.

- **Admin User.** The user that has administrator permissions for the Microsoft Azure Data Factory source system. The Admin User requires a password.
 - **Service Principal.** The authentication method that Microsoft Azure Data Factory uses for connectivity. Service Principal authentication requires a client secret key.
8. Based on the connection mode, configure connection properties with the values that you got from the administrator.
 9. Click **Next**.
- The **Configuration** page appears.

Step 2. Configure capabilities

When you configure the Microsoft Azure Data Factory catalog source, you define the settings for the metadata extraction capability and other optional capabilities.

The metadata extraction capability extracts source metadata from external source systems. You can also configure other capabilities that the catalog source includes.

You can save the catalog source configuration at any point after you enter the connection information. After you save the catalog source, you can choose to run the catalog source job. To run the job once, click **Run**. To run metadata extraction and other capabilities on a recurring schedule, configure schedules on the **Schedule** tab.

Configure metadata extraction

When you configure the Microsoft Azure Data Factory catalog source, you choose a runtime environment, define filters, and enter configuration parameters for metadata extraction.

Before you configure metadata extraction, configure runtime environments in the Informatica Intelligent Cloud Services Administrator.

1. In the **Connection and Runtime** area, choose a serverless runtime environment or the Secure Agent group where you want to run catalog source jobs.
Note: Serverless runtime environment options are available if the catalog source works with a serverless runtime environment.
2. Choose to retain, delete, or deprecate objects that are deleted from the source system in the catalog with the **Metadata Change Option**.
 - **Retain.** Retains objects that are deleted from the source system in the catalog. If you update or add a filter, the catalog retains objects extracted from the previous job and extracts additional objects that match the current filter. Objects deleted from the source system are not deleted from the catalog. Enrichments added on deleted objects and relationships are retained.
 - **Delete.** Deletes metadata from the catalog based on objects deleted from the source system and changes you make to the filter. Enrichments added on deleted objects and relationships are also permanently lost. Objects renamed in the source system are removed and recreated in the catalog.

- **Deprecate.** The lifecycle of objects imported into the catalog moves to Obsolete based on objects deleted from the source system and changes you make to the filter. This does not impact enrichments added on deprecated objects and relationships. Objects renamed in the source system are removed and recreated in the catalog. When you run the catalog source job again for other capabilities such as data classification, relationship discovery, or glossary association, the job doesn't consider obsolete objects. Obsolete objects remain in the catalog until they are purged when you run a **Purge Obsolete Objects** job on the **Explore** page.

Note: You can also change the configured metadata change option when you run a catalog source.

3. In the **Filters** area, define one or more filter conditions to apply for metadata extraction:
 - a. Select **Yes** to view filter options.
 - b. From the **Include/Exclude** list, choose to include metadata based on the filter parameters.
 - c. From the **Object type** list, select **Path**.
 - d. From the **Filter criteria** list, select **Pattern**.
 - e. Type the path to the pipeline or other asset that contains metadata that you want to include or exclude.
 - f. Click **OK**.

The following image shows the **Filter Conditions** option:

Filters

Specify metadata filters: ☐ No ☒ Yes

[Show supported wildcards and examples](#)

| | | | | | | |
|-----------------|-------------|-----------------|--|--------|---|----|
| Include/Exclude | Object type | Filter criteria | | Select | + | 🗑️ |
|-----------------|-------------|-----------------|--|--------|---|----|

4. Optionally, to define an additional filter with an OR condition, click the plus icon.
5. In the **Configuration Parameters** area, enter configuration parameters.

The following table describes the parameters that you can configure:

| Property | Description |
|--|---|
| Operational Metadata Config | <p>Specifies whether to process operational metadata.</p> <p>To process operational metadata, set the Should process operational metadata parameter to Yes.</p> <p>Configure the following parameters:</p> <ul style="list-style-type: none"> - Operational Metadata Time Range Mode. Determines the time range from which pipeline runs are processed. - Should process the latest operational metadata run only. Determines whether to process only the latest pipeline runs or all pipeline runs. <p>If you select No and run the job with the 'Delete' Metadata Change Option selected, any metadata extracted previously through the operational metadata in the pipeline instance is deleted.</p> |
| Databricks parameters | <p>Specifies the parameters for Databricks notebooks.</p> <p>Configure the following parameters:</p> <ul style="list-style-type: none"> - Notebooks Python Modules Path. The path to the directory containing custom python user modules. This path must be accessible by the Secure Agent. Example values: <code>Y:\etc\user_modules/</code> <code>opt/etc/user_modules</code> This parameter appears when you click Show Advanced. - Python Default Variables Values. Python default variable values for Databricks notebooks. Use [VARIABLES], [FUNCTIONS], or [GLOBAL] sections in the form of [SECTION NAME] before you specify a variable. This parameter appears when you click Show Advanced. - Databricks Connection. The connection details for Databricks notebooks. For more information, see "Databricks connection" on page 21. |
| SecureString entries as key-value pairs | <p>The SecureString key and value that Microsoft Azure Data Factory uses to connect to other data sources. Contact your administrator for the key and value.</p> |
| SSISDB SQL Server Connection Configuration | <p>Specifies SSISDB SQL Server connections assigned to Integration Runtimes. In the Name field, enter the name of the Integration Runtime configured to run the Execute SSIS Package activity in Microsoft Azure Data Factory. In the Connection field, enter the SSISDB SQL Server connection created in the Administrator service.</p> <p>Note: This parameter appears when you click Show Advanced.</p> |
| Additional Settings | <p>Configure expert parameters to specify additional configuration options to be passed at runtime. Required if you need to troubleshoot the catalog source job.</p> <p>Caution: Use expert parameters when it is recommended by Informatica Global Customer Support.</p> <p>Note: This parameter appears when you click Show Advanced.</p> |

6. Configure additional capabilities for the catalog source by clicking on the tabs.

Filter examples

Use filters to include or exclude metadata from different elements of the Microsoft Azure Data Factory system. Filters can contain wildcards. Use an asterisk for multiple characters and a question mark for a single character. For example, A* matches A, Ab, ABC and so on; A? matches A1, Ab, and so on.

You can use the following syntax to create filters: <FactoryName>/<FolderName>|<PipelineName>/<ActivityName>

You can choose one of the following filters to enter in the Microsoft Azure Data Factory input filters parameter:

| Syntax | Action |
|--|--|
| <FactoryName>/*/* | Includes or excludes metadata from all activities, in all pipelines of a specific factory. For example, Factory1/*/* In this example, the filter includes or excludes all activities in all pipelines of the Factory1 factory. |
| <FactoryName>/*/*/* | Includes or excludes metadata from all folders, pipelines, and activities in a specific factory. For example, Factory1/*/*/* In this example, the filter includes or excludes metadata from all folders, pipelines, and activities in the Factory1 factory. |
| <FactoryName>/<FolderName1> <FolderName2> <FolderName3> <PipelineName>/* | Includes or excludes metadata from a specific pipeline in a specific subfolder of a specific factory. For example, Factory1/Folder1 Folder2 Folder3 Pipeline1/* In this example, the pipeline name is Pipeline1. The pipeline exists in the Factory1 factory, in the Folder3 subfolder of the Folder2 folder. The Folder2 folder is in the Folder1 folder. |
| <FactoryName1>/<FolderName1> <PipelineName1>/*<FactoryName2>/<FolderName2> <PipelineName2>/* | Includes or excludes all activities in specific pipelines, folders and factories. For example, AzureDataFactory-Demo/cdgc_pipelines pl_TO_DataLake_Landing/* and AzureDataFactory-Demo/cdgc_pipelines pl_TO_DWH/* In this example, the pipeline names are pl_TO_DataLake_Landing and pl_TO_DWH. The pipelines exist in the AzureDataFactory-Demo factory, in the cdgc_pipelines folder. Note: If your pipelines are directly under a factory and not in folders, use the following syntax: <FactoryName>/<PipelineName>/<ActivityName> |
| <FactoryName>/<PipelineName>/* | Includes or excludes all activities in the specific pipeline and factory. For example, AzureDataFactory-Demo/pl_myfirstpipeline/* In this example, the filter includes or excludes all activities in the pl_myfirstpipeline pipeline in the AzureDataFactory-Demo factory. |

Python default variables values

Provide Python default variables values when your script uses values that are not defined in the code.

Ensure that the Python default variables values include either both `VARIABLES` and `FUNCTIONS` sections or a `GLOBAL` section.

To escape special characters such as `\n` or `\t`, use the backslash (`\`). For example, to define `E:\file\tgtParameterized.csv`, enter `E:\\file\\tgtParameterized.csv`

Variables

Use the following syntax: `<VariableName>=<VariableExpression>`

Note: A variable name doesn't require quotes if it contains only standard alphanumeric characters. If a variable name contains special characters, use double quotes.

A sample variable section can have the following structure:

```
[VARIABLES]
a = 42
b = 7
c = a < b ? a + 7 : b - 7 // It's 49
d = e(1,2) + 1 // Expression use call to function.
"User::table" = 'table' || "User::tableSuffix" // It's a string table concatenated with
the value of User::tableSuffix variable
```

Functions

Use the following syntax: `<FunctionCallSignature>=<FunctionExpression>`

Consider the following rules and guidelines when you define functions:

- A function name doesn't require quotes if it contains only standard alphanumeric characters.
- Enclose a function name in double quotes if it contains special characters.
- Define the list of arguments within parentheses.
- Use a question mark for each function argument.
- Ensure that arguments consist of question marks separated by commas.

A sample function section can have the following structure:

```
[FUNCTIONS]
a(?) = 1
a(?,?) = 2
b(?) = a(1) + 1 // Expression use call to another function.
c(?,?) = d + 2 // Expression use reference to variable.
```

You can provide additional sections to match functions. To match an overloaded function, provide placeholders for its arguments. You can also reference matched function arguments inside the matched section.

A sample custom function section can have the following structure:

```
[host.db.schema.func(x)]
z=x
[host.db.schema.func(x,y)]
z=x+y
```

The following table describes the functions that you can use:

| Function | Description |
|---|--|
| Hash(str, maxOutputLen) | <p>Applies the Message Digest Algorithm 5 (MD5) to an input string and produces a hash value. You can specify the length of the hash value.</p> <p>For example: <code>Hash('abcdefgh', 4) -> 'E8DC'</code></p> <p>'E8DC' is the result that you get in MD5 hashing algorithm application for the input string 'abcdefgh' with a specified hash length of 4 characters.</p> |
| Replace(str, from, to) | <p>Replaces a string with another string. For example: <code>Replace('abc', 'b', 'D') -> 'aDc'</code></p> <p>The function replaces the single occurrence of the substring 'b' in the input string 'abc' with the string 'D' and creates the modified string 'aDc'.</p> |
| ReplaceRegexp(str, regex, replacement) | <p>Replaces the strings that you specify with Java regular expressions. For example:</p> <ul style="list-style-type: none"> <code>ReplaceRegexp('abcde', 'b.*', 'f') -> 'af'</code> <code>ReplaceRegexp('graph_id20', 'id(\d+)', '\$1') -> 'graph_20'</code> <p>In the first example, the substring 'bcde' matches the regular expression pattern 'b.*'. The function replaces the matched substring with the replacement string 'f'. As a result, the modified string returned by the function is 'af'.</p> <p>In the second example the substring 'id20' matches the regular expression pattern 'id(\d+)'. The function replaces the entire matched substring with the captured digits '20'. As a result, the modified string returned by the function is 'graph_20'.</p> |
| StringLengthLimit(str, limit, hashSize) | <p>Limits the length of the input string based on a specified limit. If the length of the string exceeds the limit, the function appends a hash of the remaining characters using the Hash function, where 'hashSize' specifies the size of the hash.</p> <p>For example:</p> <ul style="list-style-type: none"> <code>StringLengthLimit('abcc', 3, 2) -> 'a26'</code> <code>StringLengthLimit('abcc', 4, 2) -> 'abcc'</code> <p>In the first example, the input string 'abcc' is longer than the specified limit of 3 characters and it is truncated to 'a'. The remaining characters are replaced with the hash value '26'.</p> <p>In the second example, the function does not modify or truncate the input string because its length matches the specified limit of 4 characters. The function returns the original string 'abcc'.</p> |
| StringLengthLimit(str, limit) | <p>Limits the length of the input string based on the specified limit. If the length of the string exceeds the limit, the function appends a hash of the remaining characters using the Hash function with a default hash size of 8 characters.</p> <p>For example: <code>StringLengthLimit('abcdabcdabcd', 10) -> 'abE340600C'</code></p> <p>In this example, the function limits the length of the input string 'abcdabcdabcd' based on the specified limit of 10 characters.</p> |
| RegexpMatch(pattern, str) | <p>Tests whether the input string matches a specified pattern. For example:</p> <p><code>RegexpMatch('[A-Za-z]+', 'Abcd') -> TRUE</code></p> <p>In this example, the function returns 'TRUE' because the input string 'Abcd' contains alphabetic characters that satisfy the pattern of one or more occurrences of alphabetic characters specified by [A-Za-z]+.</p> |
| Upper(str) | <p>Converts the characters in a given input string to uppercase.</p> <p>For example: <code>Upper('Abc') -> 'ABC'</code></p> |
| Lower(str) | <p>Converts the characters in a given input string to lowercase.</p> <p>For example: <code>Lower('aBC') -> 'abc'</code></p> |

| Function | Description |
|---|---|
| Date(text,format) | Returns objects that represent a date in a specified format. For example: Date('2017-10-31', 'yyyy-MM-dd') Note: The function follows the conventions and patterns provided by the SimpleDateFormat class in Java 8. |
| Contains/ ContainsIgnoreCase(stack, needle) | Checks if a given "needle" string is present within a "stack" string. Examples: - Contains('abc', 'ab') -> TRUE - Contains(' abc', 'AB') -> FALSE - ContainsIgnoreCase(' abc', 'AB') -> TRUE - Contains('stack', 'needle') -> FALSE In the examples, TRUE means that the function contains a given substring and FALSE means that it doesn't. |

Global

You can use a GLOBAL section. It contains both variables and function definitions. If you use a GLOBAL section, don't add the VARIABLES or FUNCTIONS sections.

Use the following syntax: DefaultVariable = 'DefaultValue'

Note: A sample global section can have the following structure:

```
[GLOBAL]
Table = 'DefaultTable'
a = 42
b = 7
c = a < b ? a + 7 : b - 7
d = e(1,2) + 1a(?) = 1
```

Note: A variable name doesn't require quotes if it contains only standard alphanumeric characters. If a variable name contains special characters, use double quotes.

Simple expression

Use the Simple Expression Language to define a value for a variable or a function call. The Simple Expression Language allows you to use values from already defined variables and functions, as well as variables from the current job context.

You can use the following case-sensitive literals:

- Integer: 42
- String: 'str'
- String with quoted single quote: 'str\''
- Boolean: true, false

You can use the following features and functionalities of the Simple Expression Language:

- Equals: 42 == 7 -> FALSE
- Non equals: 42 != 7 -> TRUE
- Concatenations: 'str1' + 'str2' -> 'str1str2'
- Function calls: Replace('abc', 'b', 'd') -> 'adc'

- Ternary operators:
 - `42 == 7 ? 'a' : 'b' -> 'b'`
 - `42 != 7 ? 'a' : 'b' -> 'a'`

Supported comment types

Use comments to provide additional information about Python default variables values. You can use the following comment types:

- Single line: `// comment`
- Multi line: `/* comment */`

Note: Comments don't affect the job.

Databricks connection

Provide connections details for Databricks notebooks.

The following table lists the parameter names and their values:

| Parameter name | Description |
|----------------|--|
| Host | The Databricks host name. |
| Workspace path | Optional. The path to the Databricks Notebook workspace. |
| Connection | The Databricks connection object that you want to use. |

| Parameter name | Description |
|--------------------------------|--|
| Catalog Preload include filter | <p>A list of include filters to preload Databricks catalog assets, such as tables or views. Use the include filter to load a limited set of assets and optimize job time. A job processes an asset when it matches at least one include filter.</p> <p>A filter value contains segments separated by periods. You can enter two wildcards in each segment. Use a question mark to represent a single character and an asterisk to represent multiple characters.</p> <p>The first two filter segments contain the Databricks catalog name and the schema name, such as <Catalog name>.<Schema name>. The next segment contains the asset name.</p> <p>The following examples illustrate the correct syntax:</p> <ul style="list-style-type: none"> - hive_metastore.schemaA.* - hive_metastore.schemaA.TableA |
| Catalog Preload exclude filter | <p>A list of exclude filters for Databricks catalog assets, such as tables or views.</p> <p>If there are no include filters, a job processes every internal Databricks asset. It does not process assets that match any of the exclude filters.</p> <p>If you specify include filters, a job processes assets that match any of the include filters and do not match any of the exclude filters.</p> <p>A filter value contains segments separated by periods. You can enter two wildcards in each segment. Use a question mark to represent a single character and an asterisk to represent multiple characters.</p> <p>The first two filter segments contain the Databricks catalog name and the schema name, such as <Catalog name>.<Schema name>. The next segment contains the asset name.</p> <p>The following examples illustrate the correct syntax:</p> <ul style="list-style-type: none"> - hive_metastore.schemaA.* - hive_metastore.schemaA.TableA |

Configure lineage discovery

Enable the lineage discovery capability and use CLAIRE to build complete lineage by recommending endpoint catalog source objects to assign to reference catalog source connections.

1. Click the **Lineage Discovery** tab.
2. Select **Enable Lineage Discovery**.
3. In the **Filters** area, define one or more filter conditions to apply for lineage discovery.

To define filters, you can choose to select catalog source types, asset groups, or enter a catalog source name or search from a list of catalog sources.

- a. Select **Yes** to view filter options.
- b. From the Include/Exclude list, choose to include or exclude catalog sources for lineage discovery based on the filter parameters.
- c. From the filter type list, select catalog source type, catalog source name, or asset group.

- d. In the filter value field, select the required catalog source types, or click the Search button and select catalog sources or asset groups.

Filters can contain the asterisk wildcard to represent multiple characters or empty text.

The following image shows the filter condition options:

Enable Lineage Discovery: ☒

Filters

Specify lineage discovery filters: ☐ No ☒ Yes

[Show supported wildcards and examples](#)

| | | | | |
|---------|---------------------|-----------------------------|---|----|
| Include | Catalog Source Type | Select Catalog Source Types | + | 🗑️ |
| Exclude | Catalog Source Name | Select Catalog Sources | + | 🗑️ |
| Exclude | Asset Group | Select Asset Groups | + | 🗑️ |

Examples:

- To include or exclude all Oracle catalog sources, select **Catalog Source Type** as the filter type and select `Oracle` in the filter value field.
- To include or exclude the 'Oracle_Retail' catalog source, select **Catalog Source Name** as the filter type and search for the catalog source or enter `Oracle_Retail` in the filter value field.
- To include or exclude all catalog sources with names that start with 'Oracle', select **Catalog Source Name** as the filter type and search for the catalog source or enter `Oracle*` in the filter value field.
- To include or exclude all catalog sources with names that end with 'Retail', select **Catalog Source Name** as the filter type and search for the catalog source or enter `*Retail` in the filter value field.
- To include or exclude all catalog sources with names that contain 'Ret', select **Catalog Source Name** as the filter type and search for the catalog source or enter `*Ret*` in the filter value field.
- To include or exclude all catalog sources that are part of the 'Financial Group' asset group, select **Asset Group** as the filter type and search `Financial Group` in the filter value field.

Note: You can't add more than one include or exclude filter for the same filter type.

- e. Optionally, to define an additional filter with an AND condition, click the **Add** icon.

For more information about lineage discovery, see *Lineage discovery* in the *Administration* help.

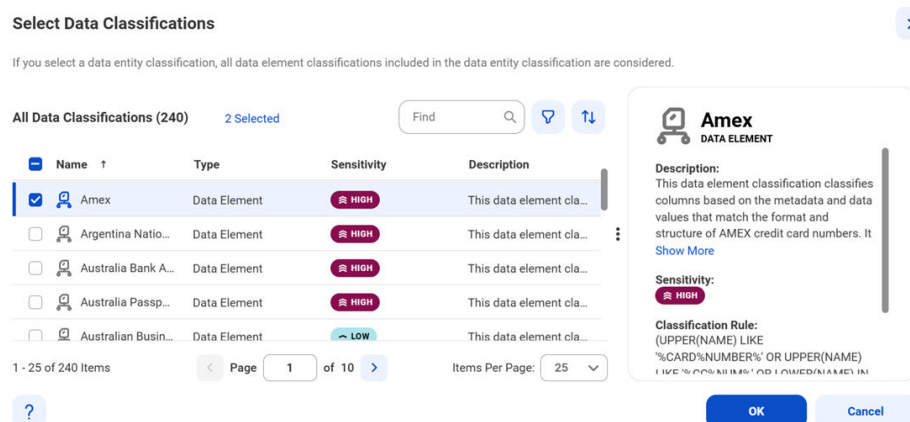
Configure data classification

Enable the data classification capability to identify and organize data into relevant categories based on the functional meaning of the data.

1. Click the **Data Classification** tab.
2. Select **Enable Data Classification**.
3. Choose one or both of the following options:
 - **Generated Data Classifications.** CLAIRE automatically generates data classifications for the data elements.

- **Data Classification Rules.** Choose from predefined or custom data classifications.

1. Click **Add Data Classification**. The following image shows the **Select Data Classifications** dialog box:



2. Select the data classifications that you want to use.
3. Click **OK**.

Configure glossary association

Enable the glossary association capability to associate glossary terms with technical assets, or to get recommendations for glossary terms that you can manually associate with technical assets in Data Governance and Catalog.

Metadata Command Center considers all published business terms in the glossary while making recommendations to associate your technical assets.

1. Click the **Glossary Association** tab.
2. Select **Enable Glossary Association**.
3. Select **Enable auto-acceptance** to automatically accept glossary association recommendations.
4. Specify the **Confidence Score Threshold for Auto-Acceptance** to set a threshold limit based on which the glossary association capability automatically accepts the recommended glossary terms.

Note: Specify a percentage from 80 to 100. If the score is higher than the specified limit, the glossary association capability automatically assigns a matching glossary term to the data element.

5. Select **Enable Below-threshold Recommendations** to receive glossary association recommendations below the auto-acceptance threshold. If you enable auto-acceptance, you can enable below-threshold recommendations to receive glossary recommendations below the auto-acceptance threshold.
6. Specify the **Confidence Score Threshold for Recommendations** to set a threshold based on which the glossary association capability makes recommendations

If you enable auto-acceptance, specify a percentage from 80 to the selected auto-acceptance threshold. You can accept or reject the recommended glossary terms that fall within this range in Data Governance and Catalog.

If you disable auto-acceptance, specify a percentage from 80 to 100 inclusive.

7. Choose to automatically assign business names and descriptions to technical assets. You can then choose to retain existing assignments and only assign business names and descriptions to assets that don't have assignments, or allow overwrite of existing assignments.

By default, existing assignments are retained.

8. Optional. Choose to ignore specific parts of data elements when making recommendations. Select **Yes** and enter prefix and suffix keyword values as needed.

Click **Select** to enter a keyword. You can enter multiple unique prefix and suffix keywords. Keyword values are case insensitive.

9. Optional. Choose specific top-level business glossary assets to associate with technical assets. Selecting a top-level asset selects its child assets as well. Select **Top-level Glossary Assets** and specify the assets on the **Select Assets** page.
10. Optional. Choose to use abbreviations and synonym definitions from lookup tables for accurate glossary association. Select **Yes** to enable, and then click **Select** to upload a lookup table.
11. Click **Next**.

The **Associations** page appears.

Step 3. Associate stakeholders and asset groups

Associate users or user groups within a stakeholder role as stakeholders for technical assets in Data Governance and Catalog. Also, you can choose to assign technical assets extracted from the catalog source to asset groups. You can then use access policies to control permissions on assets that are assigned to asset groups.

Verify that the administrator assigned users and user groups to the stakeholder role that you want to associate with technical assets.

1. To associate users or user groups as stakeholders with technical assets extracted from the catalog source, perform the following steps:
 - a. On the **Associations** page, click **Stakeholders**.
 - b. Select **Assign Stakeholders**.
 - c. Select a stakeholder role.
 - d. Click **Select** to add users and user groups from the stakeholder role as stakeholders for the technical assets.

The **Add Users & User Groups** dialog box displays a list of users and user groups assigned to the selected stakeholder role.

Add Users & User Groups

Users

User Groups

All Users (1)

Find

↕

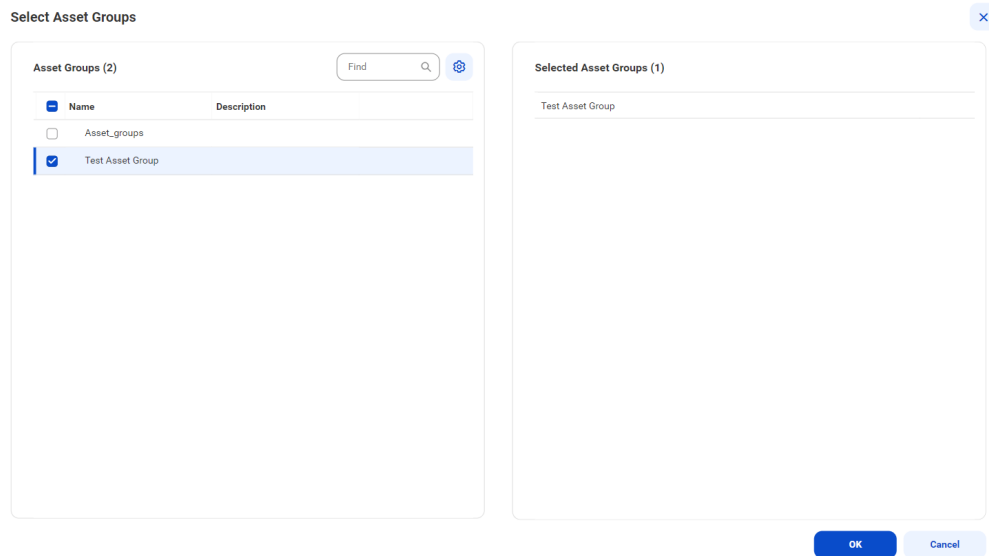
| <input type="checkbox"/> | Full Name | Email | User Name ↑ | Status |
|--------------------------|--------------|-------|-------------|--------|
| <input type="checkbox"/> | gov owner_09 | | | Active |

?

OK

Cancel

- e. Select one or more users or user groups to assign as stakeholders for the technical assets, and click **OK**.
Only the selected users and user groups belonging to the specified stakeholder role are granted the permissions to technical assets.
 - f. To assign users or user groups from another stakeholder role, click **Add** and then repeat the steps.
2. To assign asset groups to technical assets extracted from the catalog source, perform the following steps:
 - a. On the **Associations** page, click **Asset Groups**.
 - b. Select **Assign Asset Groups**.
 - c. Click **Select**.
The **Select Asset Groups** dialog box displays the list of asset groups.
If you enabled an access policy that includes an asset group, you can only view assets that belong to that asset group.
3. Select the asset groups to which you want to assign technical assets extracted from the catalog source, and click **OK**.



4. Choose to save and run the job or to schedule a recurring job.
 - To save and run the job, click **Save** and then **Run**.
 - To schedule a recurring job, click **Next** to open the **Schedule** page.

Step 4. Run or schedule the job

Choose to run a catalog source job manually, or configure it to run on schedule.

Note: You can't run multiple jobs simultaneously.

You can choose to perform a full or an incremental metadata extraction. A full metadata extraction extracts all objects from the source to the catalog. An incremental metadata extraction extracts only the changed and new objects since the last successful catalog source job run. Incremental metadata extraction doesn't remove deleted objects from the catalog and doesn't extract metadata of code-based objects if applicable.

When you run an incremental metadata extraction job with a filter to include metadata from objects, the job extracts only the objects that have the latest timestamp since the last successful job.

Note: The incremental extraction option appears if it is available for the catalog source.

Run the job manually

Click **Save** to save the catalog source and click **Run**. On the **Run Catalog Source Job** window, click **Run** to run the job.

You can override the capabilities that you selected while configuring your catalog source on the **Configuration** page. The first time you run the catalog source job, the metadata extraction capability is mandatory. From the second run onwards, you can choose to override the configured metadata change option. You can retain, delete, or deprecate objects that are deleted from the source in the catalog. For subsequent runs of the catalog source job, the metadata extraction capability is optional.

Note: You can choose incremental metadata extraction for subsequent runs only after one full metadata extraction job completes successfully. Incremental metadata extraction jobs run with the **Retain** metadata change option even if you set the option to **Delete** or **Deprecate** in the catalog source.

Note: To run a catalog source job, you need permissions on the connection to the source system. To run a catalog source job for catalog sources that reference other source systems, you need permissions on the connections for all the reference source systems.

Run the job on a schedule

You can choose to run metadata extraction and other capabilities on a recurring schedule. You can't choose incremental metadata extraction and full metadata extraction in the same schedule. To create a schedule for incremental metadata extraction, you must have completed at least one full metadata extraction job successfully. If not, first create a schedule for a full metadata extraction.

If an incremental metadata extraction is scheduled to run when the last run details aren't available, the job first performs a full metadata extraction, followed by incremental metadata extraction on subsequent runs.

For example, this can happen in the following scenarios:

- You create schedules for both incremental metadata extraction and full metadata extraction, but schedule the incremental extraction to run before the first full metadata extraction job.
- You create schedules for both incremental metadata extraction and full metadata extraction, but delete the full metadata extraction schedule before its first run.

1. On the **Schedule** tab, select **Run on Schedule**.
The **Schedule** configuration page opens.
2. Click the checkbox corresponding to each capability that you want to include in the schedule.
3. Enter the start date, time zone, and the interval at which you want to run the job.
4. You can manage additional schedules using the following options:
 - To create a new schedule, click the **Add** button.
 - To delete a schedule, click the **Delete** button.
 - To enable or disable a schedule, click the **Enable Schedule** toggle button.

Note: You can create a maximum of one schedule per capability that you enable. If you purged a catalog source or did not run the metadata extraction job, the catalog source job runs metadata extraction before running other scheduled capabilities.

Note: To create a schedule, you need permissions on the connection to the source system. If you lose permissions on the connection after you create a schedule, the scheduled jobs continue to run.

5. Click **Save** to save the schedule.

Monitor job status

After the job runs, you can monitor the status of the job on the **Overview** page of the job.

For more information about job monitoring, see *Administration*.

Step 5. Assign reference catalog source connections to endpoint catalog source objects

When you run the catalog source job, if the catalog source references another source system, a reference catalog source and connection get created that point to the reference source system. To view the complete lineage for your catalog source, you can perform connection assignment from the reference catalog source connection to the objects in the reference source system. A reference source system might be a database,

such as Databricks Notebook. You must first create and run an endpoint catalog source that connects to the reference source system.

Before you assign a connection, ensure that you have created and run an endpoint catalog source for each reference source system.

Note: If the source schema contains case-sensitive tables or if the reference objects contain multiple objects with the same name in different cases, perform case-sensitive connection assignment to get correct lineage.

If you enabled the lineage discovery capability for your catalog source, you can either curate the CLAIRE recommended endpoint objects on the **Related Catalog Sources** tab or assign connections manually.

For more information about related catalog sources and lineage discovery, see *Lineage discovery* in the *Administration* help.

1. On the **Configure** page, select the **Lineage** tab, and then select the **Lineage Discovery** tab. On the **Catalog Sources** panel, select the required catalog source and click the **Assign Connections** tab.

The **Assign Connections** tab displays a list of assigned and unassigned connections along with details for each connection. Use filters to view the connections based on the connection names. Click the **Add Filter** menu to add filters.

2. Select the connection to the reference source system and click **Assign**.

The connection name appears prefixed to the reference catalog source name on the **Hierarchy** tab of your catalog source in Data Governance and Catalog.

The **Assign Connection** dialog box appears with a list of recommended objects from the endpoint catalog sources. Click **All** to view all endpoint catalog source objects.

3. In the **Assign Connection** dialog box, select one or more objects from the endpoint catalog sources and click **Assign**.

You can filter the list in the **Assign Connection** dialog box by name, type, or endpoint.

When you click **Assign**, Metadata Command Center creates links between matching objects in the connected catalog sources, and it calculates the percentage of matched and unmatched objects. The higher the percentage of matched objects, the more accurate the lineage that you view in Data Governance and Catalog.

You can connect to the following source systems:

- Oracle. The catalog source must belong to the Database class type.
 - Microsoft SQL Server. The catalog source must belong to the Database class type.
 - Microsoft Azure Synapse Data Warehouse. The catalog source must belong to the Database class type.
 - Snowflake. The catalog source must belong to the Database class type.
 - Amazon S3. The catalog source must belong to the S3 Bucket class type.
 - Microsoft Azure Data Lake Storage Gen2. The catalog source must belong to the Microsoft Azure Data Lake Storage Container class type.
 - Databricks Notebook . The catalog source must belong to the Database class type.
 - Salesforce. The catalog source must belong to the Application class type.
 - SAP HANA Database. The catalog source must belong to the Database class type.
4. Run the catalog source job again. If you configured the catalog source job to run on a regular schedule, the next scheduled run picks up the updated details. If you didn't configure a schedule, run the catalog source job again to view complete lineage.

CHAPTER 4

View results in Data Governance and Catalog

After Metadata Command Center runs a job, you can view the results in Data Governance and Catalog where the catalog source and its elements are called technical assets. You can view a catalog source as a hierarchy. Expand each technical asset to see its components.

When referenced source systems are connected to a catalog source, you can expand the hierarchy to see details about the technical asset's component elements.

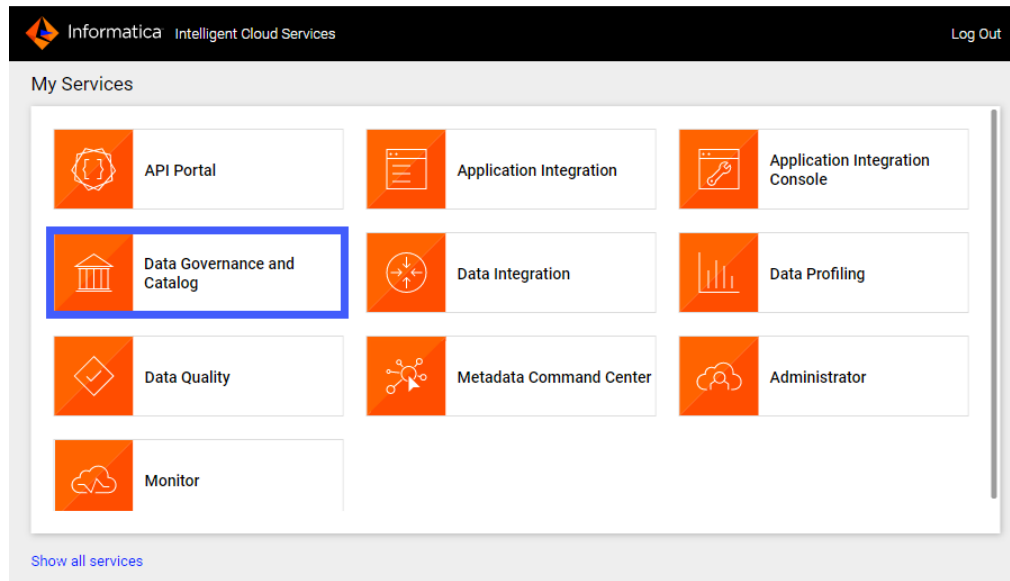
You can view the data lineage of an asset contained within a catalog source to see individual elements such as data sources, calculations, and filters. When you view data lineage, you can see the individual upstream elements that contribute data or expressions to each component of a data flow or catalog source.

View metadata extraction results

After a job runs in Metadata Command Center, view the results in Data Governance and Catalog. You can view details about source system contents as hierarchical displays and trace data lineage across task flows.

1. Log in to Informatica Intelligent Cloud Services and select Data Governance and Catalog from the **My Services** page.

The following image shows the **My Services** screen:



2. On the Data Governance and Catalog **Home** page, click the number displayed in the **Technical Assets** panel.

The **Search Results** page opens to a list of technical assets.

3. Select **Catalog Source** in the **Filters** panel.

The list of catalog sources opens. The following image shows some of the catalog sources:

Informatica Data Governance and Catalog

Search

Basic

* × Q

Filters

Clear All

Asset Type

Clear

☒ Catalog Source 214
 ☐ Column 8516
 ☐ Hierarchical field 4432
 ☐ Flat Field 2708
 ☐ Table 1965
 [Show more](#)

Lifecycle

☐ Published 214

⋮

Search Results (214)

| Name | Type | Description | Hierarchy |
|--------------------------------|--------------|-------------|-----------|
| SQL_ALL_HAWKDB1_SQL_HAWK... | Referenc... | | |
| SQL_ALL | Microsof... | | Top Level |
| rs_singlesp | Redshift ... | | Top Level |
| rs_singlesp_RSQA_ADV_RS_SP | Referenc... | | |
| sql_singlesp_HAWKDB1_SQL_HA... | Referenc... | | |
| sql_singlesp | Microsof... | | Top Level |

- Search for the catalog source from which you extracted metadata, and click on the name.
The **Overview** page of the asset opens.
The following image shows a sample asset page:

ADF_DEMO /

ADF_DEMO_adfscanner_blob_storage_adfscanner

REFERENCED CATALOG SOURCE

Overview

Hierarchy

Relationships

Stakeholders

Properties

Tickets

History

Data element 36

Data source 1

Technical Dat... 13

Data element (36)

↑↓

Find

⚙️

| Name ^ | Description | Path |
|--------------|-------------|-------------------------------------|
| categoryname | - | ADF_DEMO / ... / transactions.csv |
| categoryname | - | ADF_DEMO / ... / products.csv |
| categoryname | - | ADF_DEMO / ... / products_d.csv |
| categoryname | - | ADF_DEMO / ... / transactions_a.csv |
| customername | - | ADF_DEMO / ... / transactions_d.csv |

1 - 25 of 36 Items

<

Page 1 of 2

>

Items Per Page: 25

- View the asset from different perspectives by clicking on the tabs.

For more information about working with assets, see *Working with Assets* in Cloud Data Governance and Catalog online help.

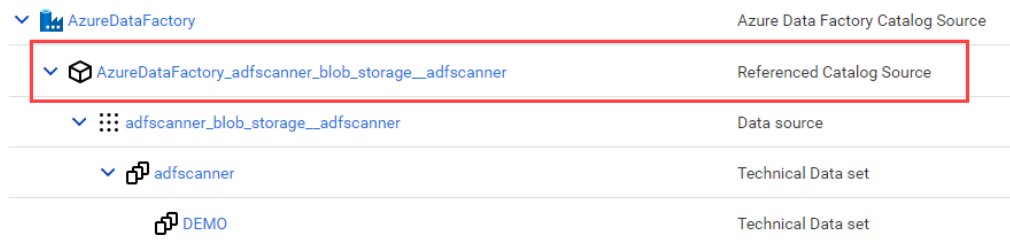
View referenced source systems

Drill down into the referenced source systems that are referenced by catalog sources.

Verify that the catalog source for the source system is assigned a connection to the catalog source for the referenced source system.

1. Browse to the catalog source that you want to view.
2. Click the Hierarchy tag to view the catalog source as a hierarchy.
3. Find an asset with the type **Referenced Catalog Source**.

The following image shows an expanded view of a referenced catalog source:



| | |
|---|-----------------------------------|
| ▼ AzureDataFactory | Azure Data Factory Catalog Source |
| ▼ AzureDataFactory_adfscanner_blob_storage_adfscanner | Referenced Catalog Source |
| ▼ adfscanner_blob_storage_adfscanner | Data source |
| ▼ adfscanner | Technical Data set |
| DEMO | Technical Data set |

4. Click to expand the referenced source to see its components.

View data lineage

Data lineage is a visual representation of the flow of data across the systems in your organization. Lineage depicts how the data flows from the system of its origin to the system of its destination.

Data lineage views are available for technical assets in the catalog source. You can view lineage at the catalog source, data set, or data element level.

The lineage at the catalog source level shows how data flows from one catalog source to another. The lineage at the data set and the data element levels show how other technical assets such as files or tables contribute to the selected asset.

If linking catalog sources is available for your catalog source, you can use Metadata Command Center to generate data lineage based on rules or by generating automated lineage with CLAIRE. You can choose source and target catalog sources and objects to link and generate lineage.

To determine whether linking catalog sources is available for your catalog source, navigate to the **Configuration** tab of the **Link Catalog Sources** page. The catalog source must appear in the list of source and target catalog sources.

For information about linking catalog sources, see *Link catalog sources* in the Administration help.

View lineage at the catalog source level

The catalog source level shows how data flows from one catalog source to another with the lineage aggregating data from the data set and data element levels.

To view data lineage at the catalog source level, open a technical asset, click the **Lineage** tab, and then verify that the level is set to **Catalog Source Level**.

Note: To improve wildcard lineage at the directory or file level for Microsoft Azure Data Factory assets, perform connection assignment, and run the Microsoft Azure Data Factory catalog source again. These wildcards can refer to files in Amazon S3 and Microsoft Azure Data Lake Storage Gen2.

View lineage at data set and data element levels

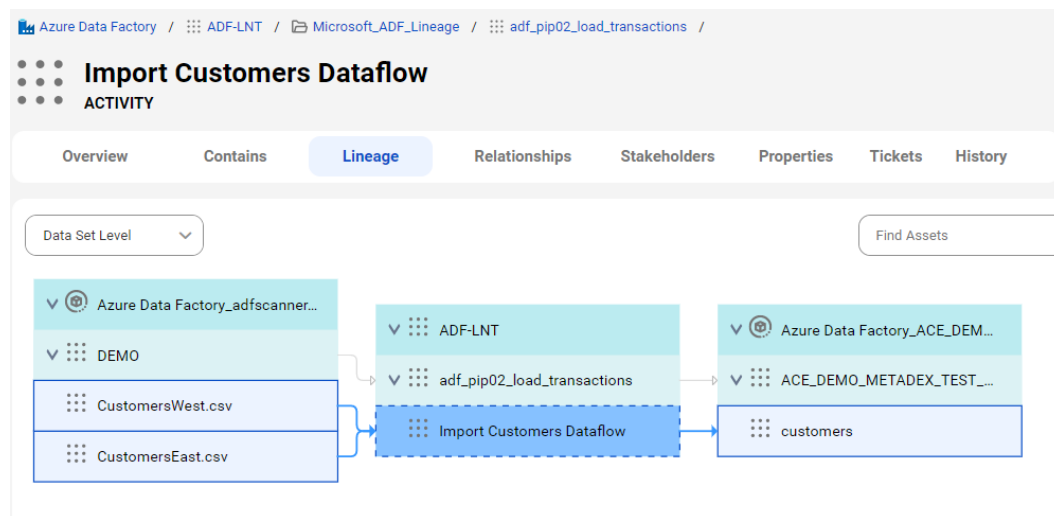
The lineage at the data set level and the data element level shows how technical assets such as files and commands contribute to the selected asset.

Data sets are technical assets that contain sets of data. Examples include files, databases, or temp files that hold the results of calculations. Data elements are objects upstream or downstream of a data set, and are accessible when you expand a data set to the data element level. For example, a column in a source object.

View lineage at the data set level

The data set level is a view that shows individual sets of data in the data flow. To view lineage at the data set level, open a technical asset, click the **Lineage** tab, and then verify that the level is set to Dataset Level.

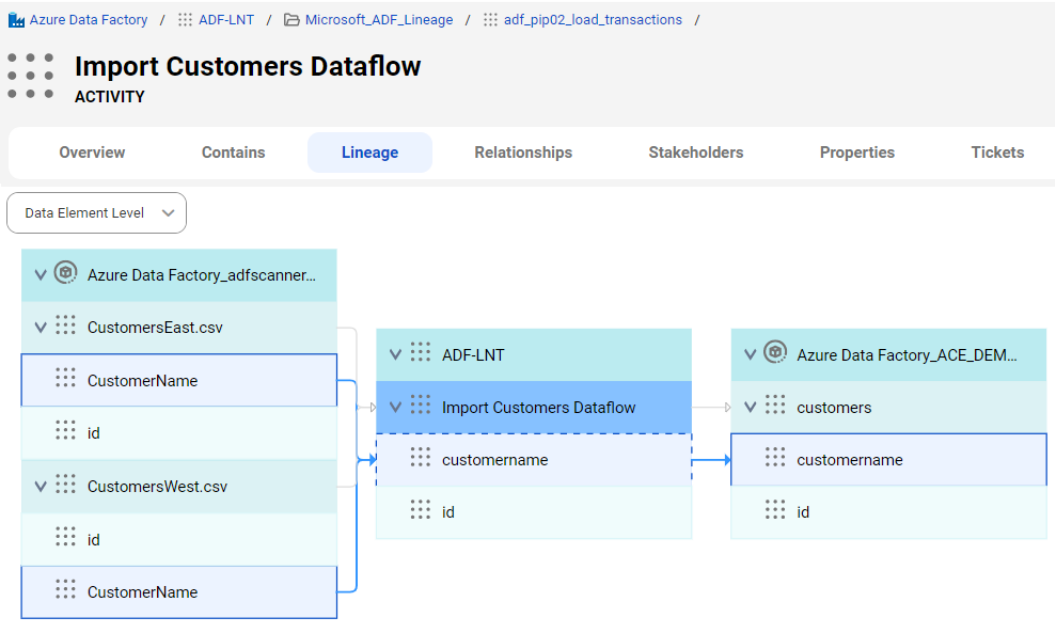
The following image shows how the Import Customers Dataflow activity gets data from two different data sets, contained in .csv files:



View lineage at the data element level

The data element level displays details of the data set level. At the data element level, you can see the input sources for expressions or commands and calculations or transformations on the data. To view data lineage at the data element level, open a technical asset, click the **Lineage** tab, and then verify that the level is set to Data Element Level.

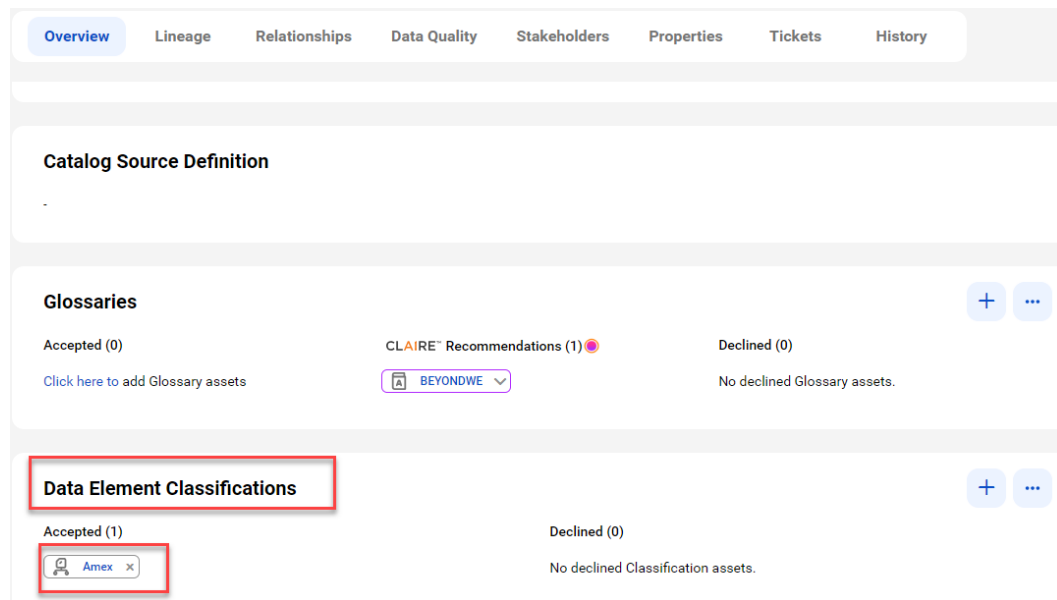
The following image shows that the Import Customers Dataflow activity uses the id and customername columns that are imported from the CustomersEast.csv and CustomersWest.csv files:



View classified data

When you add data classification rules to a catalog source in Metadata Command Center, the system identifies the columns and tables that match the rules and displays one or more matched data classifications on the column or table asset pages in Data Governance and Catalog.

The following image shows a column asset page with the inferred data element classifications that match the column data and metadata:



For more information about data classification assets, see *Asset Details* in the Data Governance and Catalog help.

View glossary associations

When you enable the glossary association capability for a catalog source in Metadata Command Center, you can view the accepted glossary assets in Data Governance and Catalog.

The **Overview** tab for a technical asset in the catalog source displays glossary assets in the Accepted and CLAIRE Recommendations sections.

The **Glossaries** panel shows the automatically accepted and CLAIRE® recommended terms.

The following image shows a sample asset page:

