Collibra

Collibra Data Intelligence Cloud

Data Lineage

Version 2022.04

Collibra

Collibra Data Intelligence Cloud - Data Lineage

Release date: April 10, 2022

Revision date: Wed Apr 06, 2022

You can find the most up-to-date technical documentation on our Documentation Center at

https://productresources.collibra.com/docs/collibra/latest/Content/to_data-lineage.htm

Contents

Collibra Data Lineage

Collibra Data Lineage is a product that allows you to trace how data flows from source to destination. It consists of two components to accommodate two different personas:

- A technical lineage for Data Engineers, Data Architects and similar personas.
- A diagram with Business Summary Lineage for Business Analysts and other business users.

Technical lineage is a detailed lineage graph that shows where data objects are used and how they are transformed. A diagram with the Business Summary Lineage shows the relations between Data Assets in Data Catalog after stitching. Both map the flow of data, but a technical lineage provides a detailed overview of the data flow, while a diagram with Business Summary Lineage only provides a summary of it.

> Note Collibra Data Lineage is only a cloud-only feature.

Technical lineage

Technical lineage is a detailed lineage graph that shows how data transforms and flows from source to destination across its entire lifecycle. It enables you to easily discover where tables and columns are used and how they relate to each other.

During the technical lineage process, relations of the type "Data Element targets / sources Data Element" are automatically created:

- Between data objects in your data source and assets from registered data sources.
- Between ingested assets from BI sources and Data Catalog assets from registered data sources.

For complete information on technical lineage, see the Collibra Data Intelligence Cloud User Guide.

About technical lineage

Technical lineage is a detailed lineage graph that shows how data transforms and flows from source to destination across its entire lifecycle. It enables you to easily discover where tables and columns are used and how they relate to each other. You use it to visualize dependencies between Table assets, Column assets, Power BI Column assets, Looker Look assets and other data objects.

During the technical lineage process, relations of the type "Data Element targets / sources Data Element" are automatically created:

- Between data objects in your data source and assets from registered data sources.
- Between ingested assets from BI sources and Data Catalog assets from registered data sources.

> Tip For complete information on ingesting metadata from the following BI tools and creating a technical lineage, see the dedicated sections:
>
> - Tableau:
> - Via the Data Catalog user interface.
> - Via the lineage harvester.
> - Looker
> - Power BI

Steps to create a technical lineage

The following table shows which steps you have to take to create a technical lineage and which prerequisites you need to execute each step.

Step	What?	Description	Prerequisites

1	Prepare Data Catalog physical data layer	Before you create a technical lineage, you prepare Data Catalog's physical data layer. This is necessary to automatically stitch assets in Data Catalog and the data elements in the data source for which you want to create a technical lineage.
		
		By preparing Data Catalog's physical data layer, you create assets of the following types:
		
		• System
		• Database
		• Schema
		• Table
		
		> Note If you don't prepare the Data Catalog physical data layer, you can still create a technical lineage. However, stitching will not be performed.
			• You have a global role with the Catalog global permission, for example Catalog Author.
			• You have a resource role with the following resource permissions:
			  ◦ Asset: Add
			  ◦ Attribute: Add
			  ◦ Domain: Add
			  ◦ Attachment: Add

Step	What?	Description	Prerequisites
2	Set up the lineage harvester	You use the lineage harvester to collect source code from your data sources and create new relations between data elements from your data source and existing assets into Data Catalog. You can download the lineage harvester from the Collibra Community Downloads page.	• Java Runtime Environment version 11 or newer or OpenJDK 11 or newer. • You have purchased Collibra Data Lineage. • You have Collibra Data Intelligence Cloud 5.7.3 or newer. • Your environment meets the hardware requirements to install and use the lineage harvester. • You have added Firewall rules so that the lineage harvester can connect to: ◦ All Collibra Data Lineage servers within your geographical location: ◦ 18.198.89.106 (techlin-aws-eu) ◦ 54.242.194.190 (techlin-aws-us) ◦ 15.222.200.199 (techlin-aws-ca) ◦ 35.205.146.124 (techlin-gcp-eu) ◦ 34.73.33.120 (techlin-gcp-us) ◦ 35.197.182.41 (techlin-gcp-au)

Step	What?	Description	Prerequisites
			◦ 34.152.20.240 (techlin-gcp-ca) ◦ 51.105.241.132 (techlin-azure-eu) ◦ 20.102.44.39 (techlin-azure-us) ◦ The host names of all databases in the lineage harvester configuration file.

Step	What?	Description	Prerequisites
3	Prepare the configuration file	You create a configuration file to determine for which data sources you want to create a technical lineage. The configuration file is used by the lineage harvester to extract information from data sources for which you want to create a technical lineage. Tip You can use the configuration file generator to create an example configuration file with the properties of your choosing. You can easily copy this example to your configuration file and replace the values of the properties to match your data source information. When you have created a configuration file, you can use specific commands to perform different actions on the data sources that are defined in your configuration file. For example, you use the full-sync command to upload the source code from the data sources in the configuration file to the Collibra Data Intelligence Cloud, where they are analyzed	• You have a global role that has the Manage all resources global permission. • You have a global role with the Catalog global permission, for example Catalog Author. • You have the Technical lineage global permission. • The lineage harvester is able to access all data sources in the configuration file. • You have the necessary permissions to all database objects that the lineage harvester accesses.

Step	What?	Description	Prerequisites
		and processed and where the technical lineage is created. Tip • If you want to use SQL files from a previously loaded data source, you have to download the SQL files of a data source to the lineage harvester. • If you want to use a data source in an external directory, for example Informatica PowerCenter, SQL Server Integration Services or IBM InfoSphere DataStage, you have to prepare the external directory folder. • If you want to use a JSON file to create a custom technical lineage, you have to prepare the JSON file.	

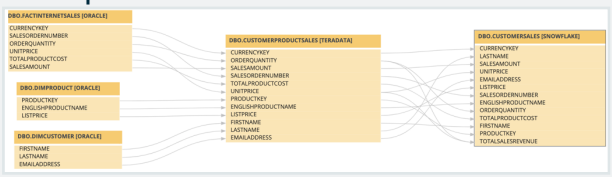
Step	What?	Description	Prerequisites
4	View the technical lineage.	After you created the technical lineage, you can go to a Power BI Column, Looker Look, Column or Table asset page and click the Technical lineage tab to view the technical lineage. You can use the Browse tab pane to search for different data objects and trace their dependencies or use the Settings tab pane to edit or export the technical lineage and see the logs created by the lineage harvester.	• You have a global role with the Catalog global permission, for example Catalog Author. • You have a global role with the Technical lineage global permission.

Data objects

You can see two types of data objects in your technical lineage:

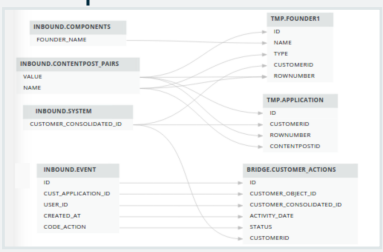
- Data objects from your data source that are stitched to assets in Data Catalog and for which you created the technical lineage. These assets have a yellow background.

  Example

- Other objects, for example temporary tables and columns, that the lineage harvester collects from your data sources, but are not stitched to assets in Data Catalog. These objects have a gray background.

Example



Warning We do not support stitching for Looker assets. We do support stitching for Power BI assets, but the stitched assets still have a gray background. This is a known issue.

Naming convention

When you create a technical lineage, Data Catalog follows a strict naming convention for the full names of assets. Each asset has a display name and full name. You can freely edit the display name. However, you should never edit the full name, because Data Catalog needs it to refresh data sources for which you created the technical lineage and to refresh the technical lineage itself.

When you prepare the Data Catalog physical data layer and the configuration file, you should always use the full name as the name of the corresponding data object in your data source for the following assets:

- Schema
- Database
- System

Note If you want to create a technical lineage for a Google BigQuery database, the project name in the configuration file must be the same as the full name of the Database asset.

Warning Editing the full name of the Schema, Database and System assets may lead to errors during the technical lineage creation process.

Transformation logic

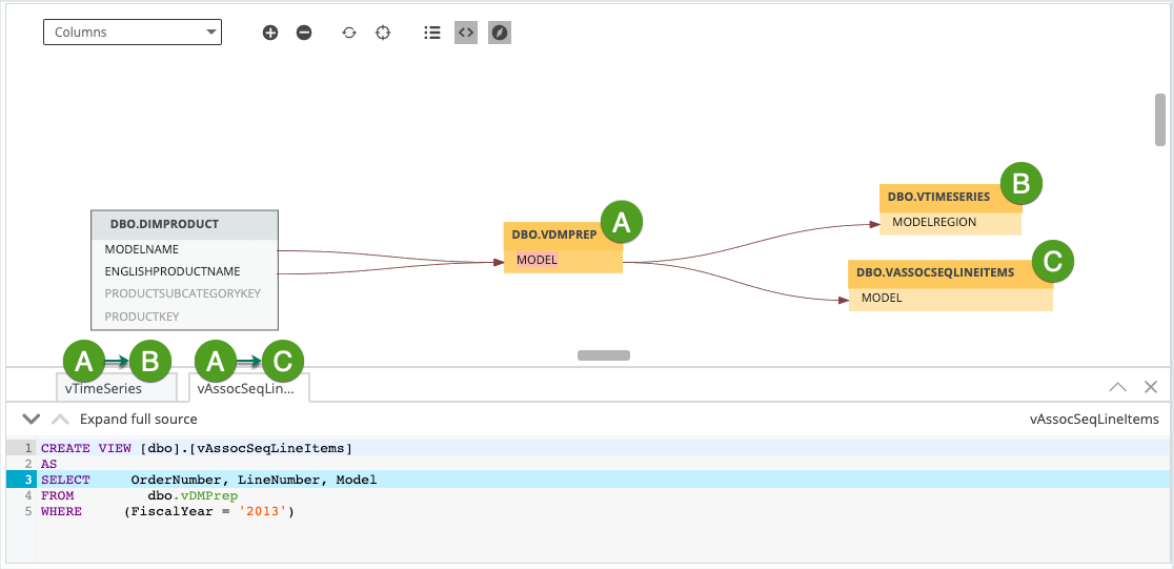
Transformation logic is used to transform source code in a technical lineage diagram that can be visualized in Data Catalog. Collibra Data Lineage supports the most commonly used transformations.

Collibra Data Lineage enables you to trace how your data flows between multiple data sources and, at the same time, see the source code of each part of your technical lineage. By following the transformations in your technical lineage, you can easily find a specific source code fragment.

Tables and columns in a technical lineage can have incoming and outgoing transformations. When you right-click on a table or column and click either Transformations (IN) or Transformations (OUT), the source code pane shows the following:

- The name of the source code fragment. On the Sources tab page, you can see the analysis log files of this source code fragment.
- If a table or column has more than one transformation, there are tabs for each source code fragment.
- The source code of the fragment. The source code that is relevant for the selected column or table is highlighted.

Example You want to see the outgoing transformations of column A to columns B and C. When you right-click column A and then click **Transformations (OUT)**, you see that there are two tabs containing source code. The first tab shows the outgoing source code from column A to column B. The second tab shows the source code from column A to column C.



Automatic stitching for technical lineage

Stitching is a process that creates relations between assets and data objects representing the same data source. More specifically, stitching creates relations between:

- the assets that were created when you prepared Data Catalog's physical data layer for a data source; and
- the data objects in the same data source for which you created a technical lineage and that represent the assets in Data Catalog.

When the data sources are scanned, the Collibra Data Lineage server automatically creates and pushes new relations of the type "Data Element targets / sources Data Element":

- Between data objects in your data source and assets from registered data sources.
- Between ingested assets from BI sources and Data Catalog assets from registered data sources.

> Note If you don't prepare the Data Catalog physical data layer, Data Catalog creates a technical lineage without stitching. As a result, when you click the Technical lineage tab on any Column, Table, Power BI Column or Looker Look asset page, you get the message **The current asset doesn't have a technical lineage yet**. However, you can use the Browse tab pane to view the technical lineage of data objects in data sources for which you created the technical lineage.

Stitching issues

To stitch assets in Data Catalog to data object collected by the lineage harvester, the Collibra Data Lineage server looks at the full path of the assets in Data Catalog and the full path of data objects in your data source. Stitching is based on the full path of objects with the following structure: (system) > database > schema > table > column. If the full paths match, the Collibra Data Lineage automatically stitches the data objects to the existing assets in Data Catalog. To indicate this, the assets have a yellow background in the technical lineage graph.

If the full path of an asset in Data Catalog does not match the full path of a data object in your data source, Collibra Data Lineage cannot stitch them. To indicate this, the data objects have a gray background in your technical lineage graph. To fix stitching issues, you must check the full path of the assets in Data Catalog and make sure they match the full path of the data objects that are shown in the technical lineage graph. If you change the full path, make sure to run the lineage harvester again.

> Warning We do not support stitching for Looker assets. We do support stitching for Power BI assets, but the stitched assets still have a gray background. This is a known issue.

> Tip You can use the Stitching tab page to easily find the full path of assets in Data Catalog and data objects that were collected by the lineage harvester. The Stitching tab page also shows an overview of all assets and data objects that are stitched successfully.

Lineage harvester versions

Collibra releases a new version of the lineage harvester every month as part of the Collibra Data Intelligence Cloud release. Check the technical lineage changelog for the most important changes in each release.

Collibra Data Intelligence Cloud version	Lineage harvester version
2022.04	2022.04
2022.03	2022.03
2022.02	2022.02
2022.01	N/A
2021.11	1.4.4
2021.10	1.4.3
2021.09	1.4.2
2021.07	1.4.1
2021.06	1.4.0
2021.05	1.3.6 1.3.5
2021.04	1.3.4

Important

- We highly recommend that you download and use the newest lineage harvester from the Collibra downloads page, even if you have an older version of Collibra Data Intelligence Cloud.
- Older lineage harvester versions are not supported.

Collibra Data Lineage servers

Collibra Data Lineage servers process and analyze the harvested metadata from supported (meta)data sources and upload it to Data Catalog. Collibra Data Lineage servers never process or store actual data, only metadata.

When you run the lineage harvester, it firsts connects to any available Collibra Data Lineage server to determine your cloud provider and geographic location of your Collibra Data Intelligence Cloud environment. Then, the lineage harvester sends the harvested metadata to the Collibra Data Lineage sever with the same cloud provider and geographic location.

Currently, your metadata can be processed on one of the following Collibra Data Lineage servers:

Server	IP address	DNS name
techlin-aws-eu	18.198.89.106	techlin-aws-eu.collibra.com
techlin-aws-us	54.242.194.190	techlin-aws-us.collibra.com
techlin-aws-ca	15.222.200.199	techlin-aws-ca.collibra.com
techlin-gcp-eu	35.205.146.124	techlin-gcp-eu.collibra.com
techlin-gcp-us	34.73.33.120	techlin-gcp-us.collibra.com
techlin-gcp-au	35.197.182.41	techlin-gcp-au.collibra.com
techlin-gcp-ca	34.152.20.240	techlin-gcp-ca.collibra.com
techlin-azure-eu	51.105.241.132	techlin-azure-eu.collibra.com
techlin-azure-us	20.102.44.39	techlin-azure-us.collibra.com

> Important You have to whitelist all Collibra Data Lineage servers in your geographic location. For example, if your data is located in Europe, you have to whitelist the following Collibra Data Lineage servers: techlin-aws-eu and techlin-gcp-eu. In addition, we highly recommend that you always whitelist the techlin-aws-us Collibra Data Lineage server as a backup, in case the lineage harvester cannot connect to other Collibra Data Lineage servers.

Supported data sources for technical lineage

Collibra Data Intelligence Cloud supports many data sources and metadata sources, including JDBC data sources, ETL tools and BI tools, for which you can create a technical lineage. You use these data sources when you prepare the configuration file and Data Catalog's physical data layer.

> Note Using an older version of a data source might not work as expected; however, we don't expect problems if you use a newer version.

JDBC data sources

The following table shows the supported JDBC data sources and driver versions that have been tested. You can connect to them via a JDBC driver or by creating a folder.

JDBC data source type	Supported versions	Connection type	Scope
Amazon Redshift	1.2.34.1058 and newer	JDBC, Folder	SQL based input without stored procedures.
Azure SQL server	Newest version	JDBC, Folder	SQL based input and stored procedures.
Azure SQL Data Warehouse	Newest version	JDBC, Folder	SQL based input and stored procedures.

JDBC data source type	Supported versions	Connection type	Scope
Azure Synapse Analytics	Newest version	JDBC, Folder	SQL based input and stored procedures.
Google BigQuery	Newest version	JDBC, Folder	SQL based input without stored procedures.
Greenplum	6.10 and newer	JDBC, Folder	SQL based input.
HiveQL (SQL-like statements)	2.3.5 and newer	Folder	SQL based input and connection via an AWS host.
IBM DB2	11.5 and newer	JDBC, Folder	SQL based input without stored procedures.
Oracle	11g, 12c and newer	JDBC, Folder	SQL based input and stored procedures.
PostgreSQL	9.4, 9.5 and newer	JDBC, Folder	SQL based input without stored procedures.
Microsoft SQL Server	2014, 2016 and newer	JDBC, Folder	SQL based input and stored procedures.
MySQL	5.7, 8 and newer	JDBC, Folder	SQL based input without stored procedures.
Netezza	7.2.1.0 and newer	JDBC, Folder	SQL based input without stored procedures.

JDBC data source type	Supported versions	Connection type	Scope
SAP Hana	2.00.40 and newer	JDBC, Folder	SQL based input and SAP HANA Information views, which includes attributes, analytic views and calculation views from database table or view data sources. Script-based calculation views and stored procedures are out of scope.
Snowflake	Newest version	JDBC, Folder	SQL based input without stored procedures.
Spark SQL	2.4.3 and newer	JDBC, Folder	SQL based input and connection via an AWS host.
Sybase Adaptive Server Enterprise	16.0 SP02 and newer	JDBC, Folder	SQL based input without stored procedures.
Teradata	15.0, 16.20.07.01 and newer	JDBC, Folder	SQL based input, including BTEQ scripts.

ETL tools

The following table shows the supported ETL tools and driver versions that have been tested. You can connect to them via an API or by creating a folder.

ETL tool	Supported versions	Connection type	Scope
AWS Glue script annotations (beta)	N/A	Folder	Only script annotations including transformation details.

ETL tool	Supported versions	Connection type	Scope
IBM InfoSphere DataStage	11.5 and newer	Folder	Commonly used DataStage ETL components including SQL overrides and transformation details. Collibra Data Lineage supports IBM InfoSphere DataStage transformation logic. You have to prepare a folder with all data objects that you want to process.
Informatica Intelligent Cloud Services, specifically Cloud Data Integration Tip Data Integration is one of the Informatica Intelligent Cloud services.	Cloud, newest only	API	Commonly used transformations in Informatica Intelligent Cloud Services: Data Integration, including SQL overrides. Supported data sources are locally stored flat files and databases.
Informatica PowerCenter	9.6 and newer	Folder	Commonly used transformations in Informatica PowerCenter, including SQL overrides. You have to prepare a folder with all data objects that you want to process.

ETL tool	Supported versions	Connection type	Scope
Matillion	Newest version	API	SQL based input without stored procedures. The lineage harvester can only access Redshift and Snowflake projects. > Note Token-based authentication is currently not supported.
SQL Server Integration Services (SSIS)	2012 and newer Package format version 6 or newer.	Folder	All commonly used transformations in SSIS, data flows and mappings, including SQL overrides. You have to prepare a folder with all data objects that you want to process.

BI tools

The following table shows the supported BI tools.

BI tool	Tested versions	Connection type
Tableau	Newest	Tableau. You have to prepare a lineage harvester configuration file for Tableau ingestion.

BI tool	Tested versions	Connection type
Power BI	Newest	Existing lineage. You have to run the Power BI harvester and the lineage harvester to ingest Power BI metadata.
Looker	Newest	Looker. You have to prepare a lineage harvester configuration file for Looker ingestion.
Power BI Report Server (beta)	SQL Server 2019	PBIRS. You have to prepare a lineage harvester configuration file for Power BI Report Server ingestion. > Note Currently, we only support Power BI Report Server ingestion in beta without stitching or technical lineage.
SQL Server Reporting Services (beta)	2021	SSRS. You have to prepare a lineage harvester configuration file for SQL Server Reporting Services ingestion.
MicroStrategy (beta)	Newest	Currently, we only support MicroStrategy ingestion without stitching or technical lineage. You can access any local or remote PostgreSQL database. The MicroStrategy Intelligence Server has an embedded PostgreSQL repository, as its default repository. For complete information on the default, embedded repository, see the MicroStrategy repository documentation.

Tip For complete information on ingesting metadata from the following BI tools and creating a technical lineage, see the dedicated sections:

- Tableau:
  - Via the Data Catalog user interface.
  - Via the lineage harvester.
- Looker
- Power BI

Custom technical lineage

You can create a custom technical lineage to include metadata of unsupported data sources. See Custom technical lineage.

Authentication

Technical lineage supports the following means of authentication:

- For all data sources, except for external directories: username and password.
- Tableau: username and password or token-based authentication.
- Google BigQuery data sources: username and password or a service account key file. For more information, see the Google BigQuery documentation.
- No other authentication methods are supported.

Lineage harvester integrations available in beta

Collibra Data Intelligence Cloud supports many data sources and metadata sources, such as ETL tools or BI sources, for which you can create a technical lineage or which you can ingest.

Before Collibra releases a new lineage harvester, we test the new lineage harvester integrations extensively. However, we cannot foresee all possible use cases and scenarios. To further improve the lineage harvester, you can now test new lineage harvester integrations in beta. After a testing period, the new lineage harvester integrations become available for all Collibra Data Lineage users

> Note Documentation is only available when the lineage harvester integrations are released. However, if you want to test new integrations, you can request testing guidelines and provide feedback.

The following table shows which integrations the lineage harvester currently supports in beta.

Metadata source	Available in lineage harvester version	Limitations	Beta process status
AWS Glue (script annotations)	1.4.0 and newer	The lineage harvester can process AWS Glue annotations in scripts coded in Python and Scala. Collibra Data Lineage does not stitch the AWS Glue metadata to Amazon S3 assets created by synchronizing an S3 File system or by registering a data source using the Collibra-provided AWS Glue driver.	Open
Power BI Report Server	1.4.0 and newer	Currently, we only support Power BI Report Server ingestion without stitching or technical lineage.	Closed
SQL Server Reporting Services	1.4.0 and newer	Currently, we only support SQL Server Reporting Services ingestion without stitching or technical lineage.	Closed

Metadata source	Available in lineage harvester version	Limitations	Beta process status
MicroStrategy	1.4.1	Currently, we only support MicroStrategy server ingestion without stitching or technical lineage. Note For the first wave of testers, we are looking for customers who use MicroStrategy on premises.	Closed

Warning The lineage harvester beta integrations offer early access to new integrations. However, we can only allow a limited number of customers to test the integrations and give feedback. We will make the integrations available for all customers after processing the feedback and improving the lineage harvester.

Testing an integration in beta

If you want to access the lineage harvester and the testing guidelines to test a lineage harvester integration in beta, do the following:

1. Create a support ticket to get access to the Technical lineage section of the Collibra Product Resources Downloads page and the testing guidelines for the lineage harvester integration.
   » You now have access to the testing guidelines.
   » You can now download and install the lineage harvester.

   Tip If you purchased Collibra Data Lineage you already have access to the newest harvester. However, you still have to create a support ticket to access the testing guidelines.
2. Test the lineage harvester integration in beta.
3. Reach out to Collibra to provide feedback via your CSM or a support ticket.

Supported SQL syntax

The SQL syntax used in your data sources has an impact on the technical lineage.

Technical lineage supports SQL syntax that is relevant to process data for all supported data sources. This includes:

- DML (Data Manipulation Language) statements that are used to move and transform data. For example, *INSERT*, *UPDATE* and *MERGE*.

  > Note Technical lineage supports the extraction of DML statements from supported procedures, but it does not support all SQL syntax.

- DDL (Data Definition Language) statements:
  - that impact the technical lineage. For example, *ALTER TABLE*, which you use to add or rename columns.
  - that are used to transform data. For example, *CREATE A TABLE AS SELECT*.
- Relevant syntax constructs. For example, nested subselects, aliases, different join methods, synonyms and cross-database references.

> Example You want to create a technical lineage for a Teradata source that has the following SQL syntax:
>
> - ALTER TYPE
> - ALTER PROCEDURE
> - CREATE/REPLACE AUTHORIZATION
> - MLOAD (MultiLoad)
> - RECORD (FastLoad)
> - BEGIN/END QUERY LOGGING
> - Functions with schema, for example schema_name.function.name(args...)
> - Functions with conversation, for example function_name(args...) RETURNS VARCHAR(<number>) CHARACTER SET LATIN
> - Macro argument attributes
>
> Collibra Data Lineage will successfully parse this SQL syntax.

Not supported SQL syntax

Technical lineage does not support the following SQL syntax:

- DML statements that you use to access data in complex structures such as JSON objects or structs.
- Triggers, foreign keys and indexes.
- Cursors, functions or dynamic queries

> Tip You can transform dynamic SQL statements into static ones. If the dynamic SQL can be logged at the runtime of a table, the dynamic query is transformed into a static query which can be extracted by the lineage harvester and processed without limitations.

Supported transformation details

Collibra Data Lineage supports the most commonly used transformations in the following sources:

- Informatica PowerCenter
- Informatica Intelligent Cloud Services
- SQL Server Integration Services
- IBM DataStage (parallel job stages)

> Note The transformation is shown if the column(expression) is using at least one column from another connected transformation.

Informatica PowerCenter transformations

The following table shows a non-exhaustive list of supported and unsupported transformations in Informatica PowerCenter.

Supported transformations	Unsupported transformations
• Aggregator • Expression • Filter • Joiner • Lookup • Mapplet • Normalizer • Rank • Sorter • Source • SQL • Stored Procedure • Target • Transaction Control	• Java • Python • XML

Informatica Intelligent Cloud Services

Collibra Data Lineage supports the following non-exhaustive list of transformations in Informatica Intelligent Cloud Services. Specifically, transformations in the Cloud Data Integration service.

- Expression
- Filter
- Joiner
- Lookup
- Mapplet
- Sequence Generator
- Source
- Target
- Union

SQL Server Integration Services (SSIS)

Collibra Data Lineage supports the following non-exhaustive list of transformations in SQL Server Integration Services:

- Aggregate
- Cache Transform
- Conditional Split
- Data Conversion
- Derived Column
- Fuzzy Grouping
- Lookup
- Merge Join
- Multicast
- OLE DB Command
- Row Count
- Script Component
- Slowly Changing Dimension
- Sort
- Union All

> Note
> - Collibra Data Lineage supports SQL, but cannot parse other languages or scripts, for example SHELL and BAT scripts.
> - All SQL queries must be preceded by the keyword SELECT, or else they will be skipped. Furthermore, if a comment precedes the keyword SELECT, the query will be skipped.

IBM DataStage

Instead of transformations, IBM DataStage uses jobs with stages. IBM Datastage has three job types: parallel jobs, sequence jobs and server jobs. Collibra Data Lineage only supports the IBM DataStage stages of parallel jobs.

For a list of all job stages per job type in IBM DataStage, read the IBM documentation.

Prepare the Data Catalog physical data layer for technical lineage

You prepare Data Catalog's physical data layer to enable Data Catalog to automatically stitch the data objects in your technical lineage to the assets in Data Catalog.

Prerequisites

- You have a global role with the Catalog global permission, for example Catalog Author.
- You have set up the JDBC driver of your source data, for example MySQL.
- You have configured one or more Jobservers in Collibra Console. If there is no available Jobserver, the **Register data source** actions will be grayed out in the global create menu of Collibra Data Intelligence Cloud.
- You have a resource role with the following resource permissions on the **Schema** community:
  - Asset > add
  - Attribute > add
  - Domain > add
  - Attachment > add
- You have the permissions to retrieve the metadata of the following database components through the JDBC Driver Database Metadata methods:
  - Schemas
  - Tables
  - Columns

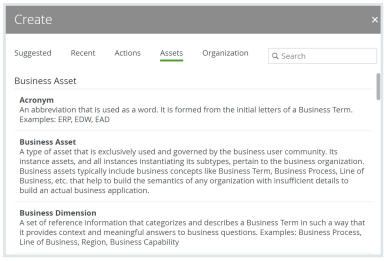
Steps

1. Create a System asset:

   > Tip The full name of your System asset must match the exact name of the system of the data source that you register in the configuration file.

   a. Open Catalog.
   b. In the main menu, click the **Create** (+) button.
      » The **Create** dialog box appears.

c. Click the **Assets** tab.



d. Click System.

» The **Create Asset** dialog box appears.

e. Enter the required information.

Field	Description
Type	The asset type of the asset that you are creating, in this case System.
Domain	The domain to which the new asset will belong. You can only create a System asset in any domain of a domain type that is assigned to a System asset type.
Name	The name of the System asset. This has to match the exact name of the system that you register in the configuration file as collibraSystemName . Tip You can create multiple assets in one go. To do this, press `Enter` after typing a value and then type the next. Depending on the settings, asset names may have to be unique in their domain. If you type a name that already exists, it will appear in strike-through style.

f. Click **Create**.

» A message at the top-right of your screen confirms that one or more assets are created.

2. Register a database as data source. You can register a database or an SQL directory as data source.

» After registration, the assets of the following asset types are created in Data

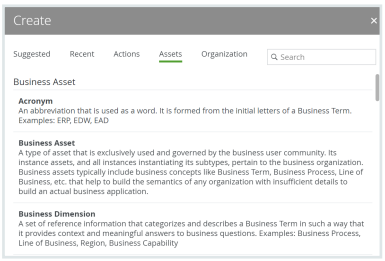
Catalog:
- Schema
- Table
- Column

> Tip The full name of your Schema asset must match the exact name of the schema in the data source that you register in the configuration file.

3. Create a Database asset:

> Tip The full name of your Database asset must match the exact name of the database or project, in case of Google BigQuery, that you register in the configuration file.

a. Open Catalog.
b. In the main menu, click the **Create** (+) button.
   » The **Create** dialog box appears.
c. Click the **Assets** tab.



d. Click Database.
   » The **Create Asset** dialog box appears.
e. Enter the required information.

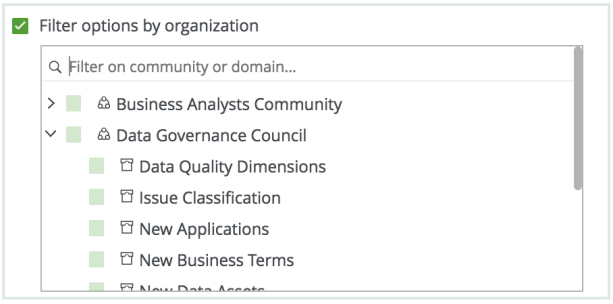
Field	Description
Type	The asset type of the asset that you are creating, in this case Database.
Domain	The domain to which the new asset will belong. You can only create a Database asset in any domain of a domain type that is assigned to a Database asset type.

Field	Description
Name	The name of the Database asset. This has to match the exact name of the database that you register in the configuration file. Tip You can create multiple assets in one go. To do this, press `Enter` after typing a value and then type the next. Depending on the settings, asset names may have to be unique in their domain. If you type a name that already exists, it will appear in strike-through style.

f. Click **Create**.
   » A message at the top-right of your screen confirms that one or more assets are created.

4. Create a relation between the System asset and the Database asset using the "Technology Asset groups / is grouped by Technology Asset" relation type.
   a. In the tab pane, click **Add Characteristic**.
      » The **Add a characteristic** dialog box appears.
   b. Click **Relations**.
   c. Search for and click **groups Technology asset**.
      » The **Add groups Technology asset** dialog box appears.

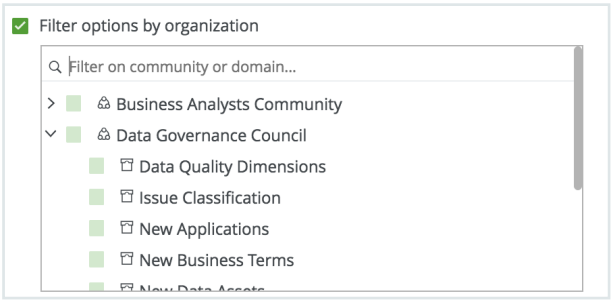
d. Enter the required information.

Option	Description
Assets	The name of the database.
Filter suggested assets by organization	Option to filter the suggestions based on selected communities and domains. If this option is selected, the organization tree appears. You can then filter and select domains and communities. 
Start date	Optionally enter the date on which the relation between the assets becomes applicable. Leave this field empty to create a permanent relation.
End date	Optionally enter the date on which the relation between the assets is no longer applicable. Leave this field empty to create a permanent relation.

e. Click **Save**.

5. Create a relation between the Database asset and the Schema asset using the "Technology Asset has / belongs to Schema" relation type.
   a. In the tab pane, click **Add Characteristic**.
      » The **Add a characteristic** dialog box appears.
   b. Click **Relations**.
   c. Search for and click **has schema**.
      » The **Add has schema** dialog box appears.

d. Enter the required information.

Option	Description
Assets	The name of the schema.
Filter suggested assets by organization	Option to filter the suggestions based on selected communities and domains. If this option is selected, the organization tree appears. You can then filter and select domains and communities. 
Start date	Optionally enter the date on which the relation between the assets becomes applicable. Leave this field empty to create a permanent relation.
End date	Optionally enter the date on which the relation between the assets is no longer applicable. Leave this field empty to create a permanent relation.

e. Click **Save**.

What's next?

If you haven't created a configuration file yet, you are now required to create it.

If you created the configuration file and prepared the physical data layer, you can run the lineage harvester to start the technical lineage process.

When the technical lineage process is finished and you have the required permissions, you can go to the asset page of a Table or Column asset from the data source that you added in the configuration file and visualize the technical lineage. At the same time, new

relations of the type "Data Element targets / sources Data Element" between assets in Data Catalog are created.

The lineage harvester also uses scheduled jobs to automate the technical lineage process.

Set up the lineage harvester

The lineage harvester is a software application that is needed to create a technical lineage and import metadata into Data Catalog.

About the lineage harvester

You use the lineage harvester to collect source code from your data sources and create new relations between data elements from your data source and existing assets into Data Catalog.

The lineage harvester runs close to the data source and can harvest transformation logic like SQL scripts and ETL scripts from a specific location, for example a database table or a folder on a file system.

The lineage harvester connects to different Collibra Data Lineage servers based on your geographical location and cloud provider. Make sure you have the correct system requirements before you run the lineage harvester. If your location or cloud provider changes, the lineage harvester rescans all your data sources.

> Note Technical lineage is created by a cloud-based environment. You only connect to the cloud via an API call that is triggered by the lineage harvester.

The lineage harvester configuration file

The lineage harvester uses a configuration file when it connects to Data Catalog via Collibra REST API. The configuration file contains references to the data sources for which you want to create a technical lineage. You have to prepare the configuration file if you want to create a technical lineage and add new relations of the type "Data Element targets / sources Data Element" between existing assets in Data Catalog and "Column is

target of / is source of Data Attribute" between assets from ingested BI sources and assets in Data Catalog.

> Warning You can only use UTF-8 or ISO-8859-1 characters in all lineage harvester files.

The lineage harvester scanners

The lineage harvester consists of many scanners that scan the data sources in your configuration file and send their metadata to the Collibra Data Lineage server. Depending on the type of data source that you want to scan, the lineage harvester uses a different scanner. Each scanner requires different properties in the lineage harvester configuration file to access your data source and scan the metadata.

Using the lineage harvester

You can use more than one lineage harvester connected to a single Collibra Data Intelligence Cloud instance, if you want to separately process data sources on different servers. In this case, all lineage harvesters must share the same configuration file, but you can determine which data sources are relevant when you run the `full-sync` command.

> Note You can use different command options and arguments that you can use to perform various actions with the lineage harvester.

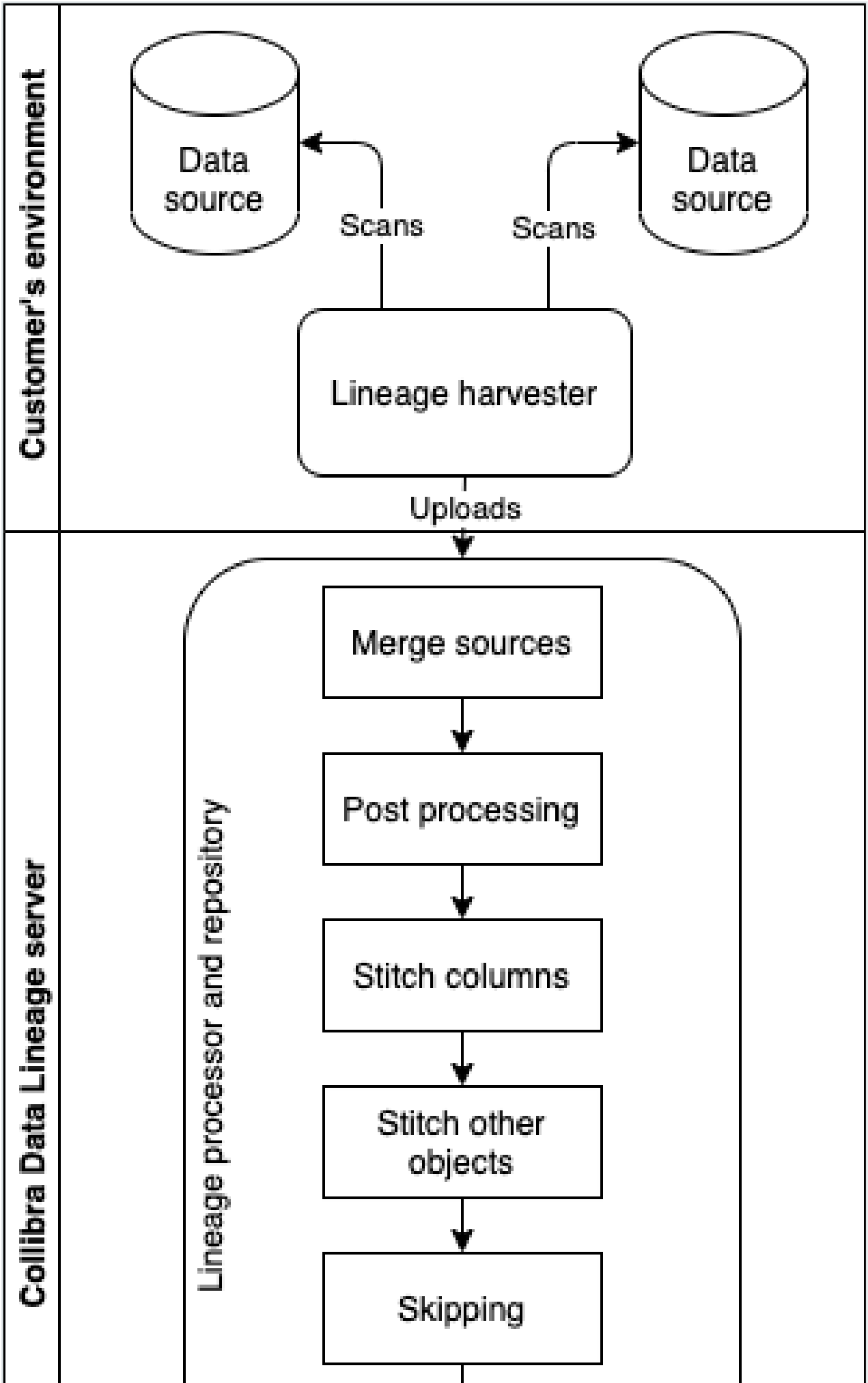
Permissions

You need a global role with the System Administration global permission, for example Sysadmin. This role must have access to all assets in the data sources in the configuration file and be able to create new relations between these assets.

Typical workflow

You use the lineage harvester to run the full-sync command. That triggers the following actions:

1. The lineage harvester:
    - Scans the data sources that are defined in the configuration file.
    - Uploads the data source information to the Collibra Data Lineage server.
2. The Lineage processor and repository on the Collibra Data Lineage server:
    - Analyzes the data sources.
    - Creates and stores the technical lineage.
    - Uploads the Column assets that exist in CollibraData Catalog.
    - Filters the results to show only relations between columns that are in Data Catalog.
3. Data Catalog:
    - Connects to the Collibra Data Lineage server to display the technical lineage.
    - Imports new relations of the type "Data Element sources / targets Data Element between existing data objects and assets of registered data sources to Data Catalog.
    - Imports new relations of the type "Column is target of / is source of Data Attribute" between BI assets and existing assets of registered data sources to Data Catalog.


Customer's environment
Data source
Scans
Scans
Data source
Lineage harvester
Uploads
Collibra Data Lineage server
Lineage processor and repository
Merge sources
Post processing
Stitch columns
Stitch other objects
Skipping

> Note The lineage harvester can only create Power BI and Looker assets if you included a reference to Power BI and Looker in the configuration file. No other assets are created during the process. Only new relations between existing or newly created Power BI and Looker assets in Data Catalog are created.

Lineage harvester system requirements

You need to meet the system requirements to be able to install and run the lineage harvester.

Software requirements

You need the following software requirements to install and run the lineage harvester.

Minimum software requirements

- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.

Recommended software requirements

- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.

Hardware requirements

You need to meet the hardware requirements to install and run the lineage harvester.

Minimum hardware requirements

You need the following minimum hardware requirements:

- 2 GB RAM
- 1 GB free disk space

Recommended hardware requirements

The minimum requirements are most likely insufficient for production environments. We recommend the following hardware requirements:

- 4 GB RAM
- 20 GB free disk space

Network requirements

You need the following minimum network requirements:

- Firewall rules so that the lineage harvester can connect to:
  - The host names of all data sources in the lineage harvester configuration file.
  - All Collibra Data Lineage servers in your geographic location:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)

    > Note The lineage harvester connects to different servers based on your geographic location and cloud provider. If your location or cloud provider changes, the lineage harvester rescans all your data sources. You have to whitelist all Collibra Data Lineage servers in your geographic location. In addition, we highly recommend that you always whitelist the techlin-aws-us server as a backup, in case the lineage harvester cannot connect to other Collibra Data Lineage servers.

> Note The lineage harvester uses port 443.

Install the lineage harvester

Before you can use the lineage harvester, you need to download it and install it. You can download the lineage harvester from the Collibra Community downloads page.

> Tip Install your lineage harvester close to your data source or on the same server.

> Tip If you only want to install the lineage harvester for:
>
> - Power BI ingestion, click here.
> - Looker ingestion, click here.

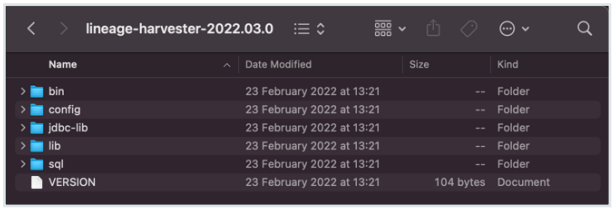
> Warning If you upgrade to lineage harvester 1.3.0 or newer, you have to follow an upgrade procedure.

Prerequisites

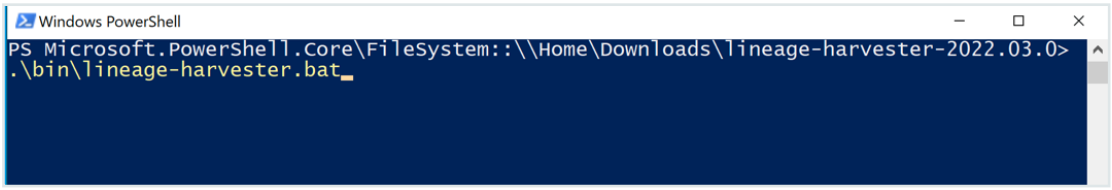
- You have purchased Collibra Data Lineage.
- You have Collibra Data Intelligence Cloud 5.7.3 or newer.
- You meet the minimum system requirements.
- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.
- You have added Firewall rules so that the lineage harvester can connect to:
  - All Collibra Data Lineage servers within your geographical location:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)
  - The host names of all databases in the lineage harvester configuration file.

Steps

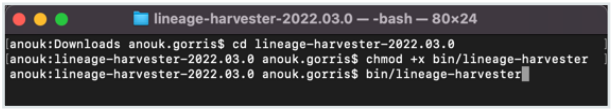
1. Download the lineage harvester.
2. Unzip the archive.
   » You can now access the lineage harvester folder.



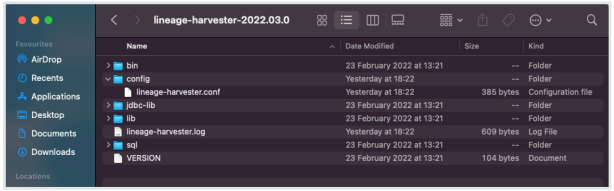
3. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



   » The lineage harvester is installed automatically. You can check the installation by running `./bin/lineage-harvester --help`.

What's next?

You can now prepare the lineage harvester configuration file and run the lineage harvester again.

Lineage harvesting app command options and arguments

After creating a configuration file, you can use the lineage harvester to perform specific actions with the data sources that are defined in your configuration file.

> Tip If you run the lineage harvester in command line, you will see an overview of possible command options and arguments that you can use. If the lineage harvester process fails, you can use the technical lineage troubleshooting guide to fix your issue.

Typical command options and arguments

The following table shows the most commonly used command options and arguments.

> Note You can use more than one lineage harvester connected to a single Collibra Data Intelligence Cloud instance, if you want to separately process data sources on different servers. In this case, all lineage harvesters must share the same configuration file, but you can determine which data sources are relevant when you run the `full-sync` command.

Command	Description
`full-sync`	Uploads all of the metadata from the data sources mentioned in your configuration file to the Collibra Data Lineage server, where the metadata is then processed and uploaded to Data Catalog.

Command	Description
`-s "<ID of data source>"`	Uploads only the metadata from a specified data source. For example, `full-sync -s "myOracleDataSource"`. The specified data source must be mentioned in your configuration file. This command allows you to process data from a newly added data source or to refresh a data source in the configuration file, without refreshing the other data sources. This reduces the time you need to upload your data sources, since you only upload specific ones without affecting the others. If you want to process multiple data sources, add `-s "ID of another data source"` per data source to the command. > Note You can use this argument multiple times to include multiple data sources.

Command	Description
`--no-matching`	Uploads a technical lineage without stitching the data objects in your technical lineage to the corresponding Column and Table assets in Data Catalog. > Note As a result, you won't see the technical lineage of a specific Table or Column asset, but you can still see and browse the full technical lineage.

Command	Description
`sync`	Whereas `full-sync` ingests metadata onto the Collibra Data Lineage server, processes the metadata and syncs it with assets in Data Catalog, the `sync` command only performs this last part: it syncs the metadata–as it exists on the Collibra Data Lineage server–and your assets in Data Catalog. Tip See the following example for advice on how to use the `sync` command to add a new data source without re-harvesting all data sources. **Example** Let's say you've run `bin/lineage-harvester full-sync`, to upload from all data sources, process the metadata and sync with Data Catalog. You then decide that you want to add a new data source, but not harvest all data sources again. 1. Reference the new data source in the lineage harvester configuration file. Let's say that the new data source has the ID "MyNewSource". 2. Run `bin/lineage-harvester load-sources -s MyNewSource`, to load the new data source and create the ZIP file. 3. Run `bin/lineage-harvester`

Command	Description
	`analyze ${zip_file_from_step_2}`, to analyze the new data source on the Collibra Data Lineage server. 4. Run `bin/lineage-harvester sync`, to sync all of the data sources referenced in your configuration file and Data Catalog.
`-s "<ID of data source>"`	Syncs only the metadata on the Collibra Data Lineage server, from a specified data source. For example, `sync -s "myOracleDataSource"`. The specified data source must be mentioned in your configuration file. This command allows you to sync data from one data source without refreshing the other data sources. You must have previously uploaded the metadata to the Collibra Data Lineage server. > Note You can use this argument multiple times to include multiple data sources.
`load-sources`	Downloads all your data sources in a separate ZIP file, per data source, to the lineage harvester output folder.

Command	Description
`-s <ID of data source>`	Downloads only the data source with a specific ID. For example, `load-sources -s "myOracleDataSource"`. Note You can use this argument multiple times to include multiple data sources.
`cat passwords.json | ./bin/lineage-harvester <command-like-full-sync> --passwords-stdin`	Provides passwords of your Collibra Data Intelligence Cloud instance and the data sources in your configuration file to the lineage harvester without storing the passwords in the lineage harvester folder. You can replace `cat passwords.json` by a string generated by your password manager.
`test-connection`	Checks the connectivity to the Collibra Data Lineage server and to Data Catalog. The logs will also show the IP addresses of the Collibra Data Lineage servers that you have to whitelist. This command is mostly used for troubleshooting purposes.
`--help`	Shows an overview of all supported command options and arguments that you can use in the lineage harvester.

Command	Description
`--version`	Shows the version of the lineage harvester that you are using.
`-Dlineage-harvester.log.dir=path/to/log/dir`	Determine the path of the log file.

Technical lineage password manager integration design

If you don't want the lineage harvester to store your passwords, you can store them in your password manager. As a result, when you run the lineage harvester, you provide your passwords in a prescribed JSON structure via stdin.

Structure of the JSON file

If you prepare a JSON file with your passwords, you have to name the file *passwords.json*.

The JSON file must have two sections:

- The "`catalogs`" section defines the connection information and credentials to your Collibra Data Intelligence Cloud instance.
- The "`sources`" section defines the connection information and credentials to your data sources. You use the same "id" as the `id` property in the lineage harvester configuration file.

The JSON file must have the following structure:

```
{
 "catalogs": [
  {
   "url" : "<url-to-collibra-cloud>",
   "username":"<username-to-sign-in-to-collibra>",
   "password": "<password-to-sign-in-to-collibra>"
  }
 ],
 "sources": [
  {
   "id": "<id-of-your-database>",
   "username": "<database-username>",
   "password": "<database-password>"
```

```
  }
 ]
}
```

Examples of commands

When you run the lineage harvester, you can use one of the following commands to provide the passwords:

Passwords location	Command
a locally stored JSON file	`cat passwords.json | ./bin/lineage-harvester full-sync --passwords-stdin`
a custom script, for example from a password manager	`<prepare-passwords-command> | ./bin/lineage-harvester full-sync --passwords-stdin` Note Depending on your password manager, you may need different parameters. For example, see the LastPass documentation for the parameters needed by LastPass.

Connecting to a proxy server

Technical lineage does not support proxy server authentication, but you can connect to a proxy server via the following commands.

On Windows

1. Set the `-D` parameter to the `JAVA_OPTS` environment variable.

   Example
   ```
   set JAVA_OPTS=-Dhttps.proxyHost="azusquid.imf.org" -Dhttps.proxyPort="8080"
   ```

2. Run the lineage harvester in the same command line window: `.\bin\lineage-harvester.bat`

On other operating systems

1. To access the hosts via a proxy server, run the following command: `bin/lineage-harvester -Dhttps.proxyHost=<Hostname or IP address of the proxy> -Dhttps.proxyPort=<port number> full-sync`

   > Example If you want to use a proxy with hostname *proxy.example.com* and port number *443*, run the following command:
   >
   > ```
   > bin/lineage-harvester -Dhttps.proxyHost=proxy.example.com
   > -Dhttps.proxyPort=443
   > ```

2. To exclude hosts that should be accessed without going through the proxy server, add the following parameter: `-Dhttp.nonProxyHosts=<host to exclude>`. You can exclude multiple hosts by using the pipe character (|) to separate the hostnames or IP addresses to exclude. You can also use an asterisk (*) as a wildcard to match multiple hostnames or IP addresses.

   > Example If you want to exclude hosts with hostname *localhost* and hosts with IP address *127.0.0.1* and all IP addresses starting with *192.168**, run the following command:
   >
   > ```
   > bin/lineage-harvester -Dhttps.proxyHost=proxy.example.com
   > -Dhttps.proxyPort=443 -
   > Dhttps.nonProxyHost=localhost|127.0.0.1|192.168*
   > ```

> Important In your configuration file, the value of the source "url" or "hostname" property (depending on the data source), and the value in your `-Dhttps.nonProxyHost` parameter, as described above, must both be either an IP address or a host name. You will get an error if, for example, you have a host name in the "hostname" property and an IP address in the `-Dhttps.nonProxyHost` parameter.

Prepare the lineage harvester configuration file

Before you can visualize the technical lineage or ingest a BI source, you have to create a configuration file for the (meta)data sources that you want to process. This configuration

file is used by the lineage harvester to extract data from (meta)data sources for which you want to create a technical lineage or you want to ingest.

Note

- Technical lineage only supports a limited list of (meta)data sources.
- You can only use UTF-8 or ISO-8859-1 characters in all lineage harvester files.
- Each data source has an ID property. The ID string must be unique and human readable. The ID can be anything and is only used to identify the batch of metadata that is processed on the Collibra Data Lineage server.
- The lineage harvester connects to different servers based on your geographical location and cloud provider. Make sure you have the correct system requirements before you run the lineage harvester. If your location or cloud provider changes, the lineage harvester rescans all your data sources.
- Technical lineage supports the following means of authentication:
  - For all data sources, except for external directories: username and password.
  - Tableau: username and password or token-based authentication.
  - Google BigQuery data sources: username and password or a service account key file. For more information, see the Google BigQuery documentation.
  - No other authentication methods are supported.
- The lineage harvester does not support proxy server authentication, but you can manually connect to a proxy server via command line. For more information, see Connecting to a proxy server.
- Comments in the lineage harvester configuration file are not supported.
- If you upgrade to lineage harvester 1.3.0 or newer, you have to follow an upgrade procedure.

Tip For complete information on ingesting metadata from the following BI tools and creating a technical lineage, see the dedicated sections:

- Tableau:
  - Via the Data Catalog user interface.
  - Via the lineage harvester.
- Looker
- Power BI

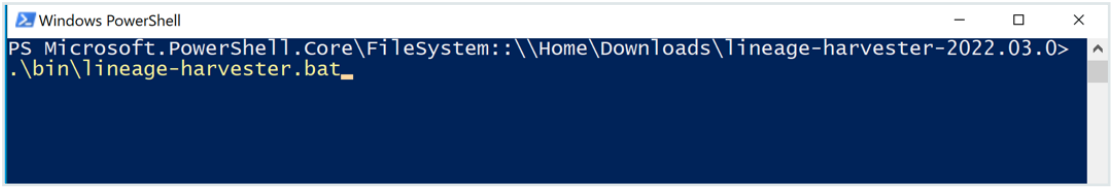
Prerequisites

- You have prepared the physical data layer in Data Catalog.
- You have a global role that has the System administration global permission.
- You have a global role that has the Manage all resources global permission.
- You have a global role with the Technical lineage global permission.
- You have downloaded the lineage harvester and you have the necessary system requirements to run it.
- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.
- You have added Firewall rules so that the lineage harvester can connect to:
  - All Collibra Data Lineage servers within your geographical location:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)
  - The host names of all databases in the lineage harvester configuration file.
- If you want to use a previously loaded data source, you have downloaded the SQL files of the data source to the lineage harvester.
- If you want to use an external directory, you have prepared a folder with data objects from the external directory.
- You have the necessary permissions to all database objects that the lineage harvester accesses.

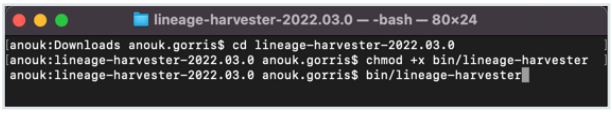
> Note For a detailed overview of the permissions that you need to access the data objects of your data sources, go to the online version of the user guide.

Steps

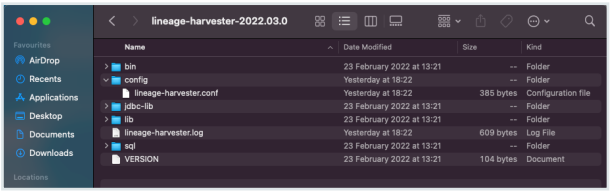
1. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



2. Open the configuration file and enter the values for each property.

> Tip You can use the configuration file generator to create an example configuration file with the properties of your choosing. You can easily copy this example to your configuration file and replace the values of the properties to match your data source information.

Properties	Description
general	This section describes the connection between Collibra lineage and Data Catalog.
catalog	This section contains information that is necessary to connect to Data Catalog. Note Versions of the lineage harvester older than 1.1.2 show `collibra` instead of `catalog`.

Properties	Description
url	The URL of your Collibra environment. > Note You can only enter the public URL of your Collibra environment. Other URLs will not be accepted.
username	The username that you use to sign in to Collibra.

Properties	Description
useCollibraSystemName	Indication whether you want to use the system or server name of a data source to match to the System asset you created when you prepared the physical data layer. This is useful when you have multiple databases with the same name. By default, the `useCollibraSystemName` property is set to `False`. If you want to use it, set it to `True`. ◦ If you keep the `useCollibraSystemName` property set to `false`, the lineage harvester ignores the `collibraSystemName` property in the rest of the configuration file. ◦ If you set the `useCollibraSystemName` property to `true`, the lineage harvester reads the value in the `collibraSystemName` property in all sections of the configuration file and in the following files: ▪ The Informatica <source ID> configuration file > Important You must prepare a <source ID> configuration file regardless of whether the `useCollibraSystemName` property in your lineage harvester configuration files is set to `true` or `false`. ▪ The IBM DataStage or SQL Server Integration Services connection definition configuration files. ▪ The Informatica Intelligent Cloud Services <source ID> configuration file. > Important You must prepare a <source ID> configuration file regardless of whether the `useCollibraSystemName` property in your lineage harvester configuration files is set to `true` or `false`.

Properties	Description
	■ The Power BI <source ID> configuration file. ■ The Looker <source ID> configuration file. ■ The JSON files with a predefined lineage. Note For SQL data sources, if the `useCollibraSystemName` property is: ◦ `false`, system or server names in table references in analyzed SQL code are ignored. This means that a table that exists in two different systems or servers is identified (either correctly or incorrectly) as a single data object, with a single asset full name. ◦ `true`, system or server names in table references are considered to be represented by different System assets in Data Catalog. The value of the `collibraSystemName` property is used as the default system or server name. Warning Unless you have multiple databases with the same name, we highly recommend that you don't change the default value.
sources	This section describes the data sources for which you want to create the technical lineage. You have to create a configuration section for each data source. Note You can add multiple data sources to the same configuration file.
<SQL directory properties>	This configuration section contains the required information of one individual SQL directory with connection type "Folder".

Properties	Description
id	The unique ID of the data source. For example, `my_first_data_source`.
type	The kind of data source. In this case, the value has to be `SqlDirectory`.
path	The full path to the SQL directory.
mask	The pattern of the file names in the directory. By default, this is `*`.
recursive	Indication of the files you want to harvest: ◦ `false` (default): Only harvest the files in directly under the folder in the SQL directory path. ◦ `true`: Harvest all files under the folder in the SQL directory path and subdirectories.

Properties	Description
dialect	The dialect of the database. Tip You can enter one of the following values: ◦ *azure*, for an Azure SQL Server data source. ◦ *bigquery*, for a Google BigQuery data source. ◦ *db2*, for an IBM DB2 data source. ◦ *hana*, for a SAP Hana data source. ◦ *hana-cviews*, for SAP Hana data calculation views. ◦ *hive*, for a HiveQL data source. ◦ *greenplum*, for a Greenplum data source. ◦ *mssql*, for a Microsoft SQL Server data source. ◦ *mysql*, for a MySQL data source. ◦ *netezza*, for a Netezza data source. ◦ *oracle*, for an Oracle data source. ◦ *postgres*, for a PostgreSQL data source. ◦ *redshift*, for an Amazon Redshift data source. ◦ *snowflake*, for a Snowflake data source. ◦ *spark*, for a Spark SQL data source. ◦ *sybase*, for a Sybase data source. ◦ *teradata*, for a Teradata data source. If you want to use a Spark SQL data source, make sure that you have an AWS host.

Properties	Description
database	The name of your database, which is the full name of your Database asset. Note You have to use the same database name as the full name of the Database asset that you create when you prepare the physical data layer in Data Catalog. Important Teradata and MySQL data sources do not have schemas. As a result, Teradata and MySQL databases are stored in Data Catalog and technical lineage as Schema assets. The technical lineage Browse tab pane shows the following names: ◦ For Teradata: ▪ The database name is the name that you enter in the `collibraSystemName` property. ▪ The schema name is the name that you enter in the `database` property. ◦ For MySQL: ▪ The database name is the name that you enter in the `database` property.
collibraSystemName	The name of the data source's system or server. This is also the full name of your System asset in Data Catalog. You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog.

Properties	Description
schema	The name of the default schema, if not specified in the data source itself. This corresponds to name of your Schema asset. Note You must use the same schema name as the name of the Schema asset that you create when you prepare the physical data layer in Data Catalog.
verbose	Indication whether you want to enable verbose logging. By default this is set to `True`. If you don't want to use verbose logging, set it to `False`.
<External directories>	This configuration section contains the required information to connect to the following data sources: ◦ Informatica PowerCenter ◦ SQL Server Integration Services (SSIS). ◦ IBM InsfoSphere DataStage Note Make sure that you have prepared a local folder with the Informatica objects, SSIS files or DataStage files for which you want to create a technical lineage.
collibraSystemName	The name of the data source's system or server. If the `useCollibraSystemName` property is set to `true`, you must prepare a configuration file to provide the system information.
id	The unique ID of your data source. For example, *my_informatica*.

Properties	Description
type	The kind of data source. In this case, the value has to be *ExternalDirectory*.
dirType	The type of external directory. The value has to be one of the following: ◦ `infa`, for an Informatica PowerCenter data source. ◦ `ssis`, for a SQL Server Integration Service data source. ◦ `datastage`, for a IBM InfoSphere DataStage source.
path	The full path to the folder where you stored the data source.
mask	The pattern of the file names in the directory. By default, this is `*`.
recursive	Indication whether you want to use recursive queries. By default, this is set to `False`. If you want to use recursive query, set it to `True`.
<Informatica Intelligent Cloud Services Data Integration>	This configuration section contains the required information to enable the lineage harvester to collect and process Data Integration objects. Tip Make sure you have READ permission on all data objects that you want to harvest.
type	The kind of data source. In this case, the value has to be `IICS`.

Properties	Description
id	The unique ID that is used to identify the data source on the Collibra Data Lineage server. For example, `my_data_integration`.
collibraSystemName	The name of the Informatica server or system. Important You must prepare a <source ID> configuration file to provide this system information. This is true regardless of whether the `useCollibraSystemName` property is set to *true* or *false*.
loginURL	The URL of the Informatica Intelligent Cloud Services environment sign-in page. For example: `https://dm-us.informaticaintelligentcloud.com`.
username	The username you use to sign in to Informatica Intelligent Cloud Services.

Properties	Description
objects	The objects that you want to export. Each object requires a path and a type, for example: `"objects": [ { "path" : "Sales", "type" : "Project" }, { "path" : "Finance/Task_Flows", "type" : "Folder" }, { "path" : "Common/Task_Flows/tf_CalendarDimension", "type" : "Taskflow" } ]` The following section provides information to identify and access Data Integration objects. Tip For more information about the objects that you can export and the required information, see the Informatica documentation.
path	The full path to the object.
type	The type of the object. For example, Taskflow. IICS scanner's starting point is a Taskflow. Therefore the only meaningful types to export are: Taskflow, Project and Folder. Note The types are not case sensitive.

Properties	Description
paramFiles	The full path to the directory in which your parameter files are stored. This is an optional parameter that allows you to harvest parameter files in Informatica Intelligent Cloud Services data sources. Important The hierarchy of the files in the directory must be an exact match of the hierarchy of the files in your file system. **Show me how to do this** a. Create a directory for your parameter files. For this example, let's name the directory *my-parameter-files*. b. In your lineage harvester configuration file, the value of the `paramFiles` property needs to be the full path to your parameter files directory, for example `/full/path/<my-parameter-files>/`. c. Copy your parameter files to your parameter files directory. Be sure to preserve the full path for each of your parameter files. For example, for parameter file */root/child/child2/paramfile.txt*, run the following commands: i. `cd /full/path/<my-parameter-files>/` ii. `mkdir -p root/child/child2/` iii. `cp /root/child/child2/paramfile.txt root/child/child2/`

Properties	Description
<Matillion>	This section contains the required information for Matillion. Tip When you create a new project in Matillion, you define in which group you want to create the project, the project name and the environment name. This information is needed to enable the lineage harvester to access Matillion and scan your metadata. Important Currently, you can only create a technical lineage for Snowflake and Redshift projects in Matillion.
id	The unique ID that is used to identify the data source on the Collibra Data Lineage server. For example, `my_matillion_data_integration`.
type	The kind of data source. In this case, the value has to be `Matillion`.
url	The URL of your Matillion environment. For example, `https://<domain name>` or `https://<IP address>`.
groupName	The name of your group in Matillion.
projectName	The name of your project in Matillion. You can only add the name of one project. If you want to create a technical lineage for other projects within the same group, create a new section in the lineage harvester configuration file.

Properties	Description
environmentName	The name of your environment in Matillion. You can only add the name of one environment. If you want to create a technical lineage for other environments within the same project, create a new section in the lineage harvester configuration file.
dialect	The dialect of the database. You can enter one of the following values: ◦ `redshift`, for an Amazon Redshift data source. ◦ `snowflake`, for a Snowflake data source.
username	The username that you use to sign in to Matillion.
startTimestamp	The timestamp of tasks in Matillion. You can use this parameter to limit the amount of metadata that the lineage harvester scans. If the `startTimestamp` field remains empty or is deleted from the configuration file, all accessible tasks are scanned. Matillion automatically removes entries older than seven days.
collibraSystemName	The name of the Matillion system or server.

Properties	Description
<Custom lineage>	This section contains the required information to connect to a custom lineage. You create a custom lineage by adding connection properties to a JSON file containing a predefined technical lineage. Make sure that you have prepared a local folder with the JSON file that contains the predefined technical lineage. > Note In the local folder that you need to create, you can only have one JSON file. You can, however, add other files in the harvested directory and subdirectories and refer to those files from within the JSON file.
id	The unique ID of your custom technical lineage. For example, `MyCustomLineage`.
type	The kind of data source. In this case, the value has to be `ExternalDirectory`.
dirType	The type of external directory. In this case, the value is `custom-lineage`.
path	The full path to the folder where you stored the data source or JSON file.
<database properties>	This configuration section contains the required information of one individual data source with connection type "JDBC".
id	The unique ID of your data source. For example, `my_second_data_source`.

Properties	Description
type	The kind of data source. In this case, the value has to be `Database`.
username	The username that you use to sign in to your data source.
dialect	The dialect of the database. Tip You can enter one of the following values: ◦ *azure*, for an Azure SQL Server data source. ◦ *db2*, for an IBM DB2 data source. ◦ *hana*, for a SAP Hana data source. ◦ *hana-cviews*, for SAP Hana data calculation views. ◦ *greenplum*, for a Greenplum data source. ◦ *mssql*, for a Microsoft SQL Server data source. ◦ *mysql*, for a MySQL data source. ◦ *netezza*, for a Netezza data source. ◦ *oracle*, for an Oracle data source. ◦ *postgres*, for a PostgreSQL data source. ◦ *redshift*, for an Amazon Redshift data source. ◦ *spark*, for a Spark SQL data source. ◦ *sybase*, for a Sybase data source. ◦ *teradata*, for a Teradata data source. If you want to use a Spark SQL data source, make sure that you have an AWS host.

Properties	Description
databaseNames	The names or IDs of your databases. Enter the database names of your data source between double quotes ("") and put everything between square brackets. If you want to include more than one database, separate them by a comma. For example, ["`MyFirstDatabase`", "`MySecondDatabase`"]. Note You have to use the same database names as the full names of the Database assets that you create when you prepare the physical data layer in Data Catalog. Important Teradata and MySQL data sources do not have schemas. As a result, Teradata and MySQL databases are stored in Data Catalog and technical lineage as Schema assets. The technical lineageBrowse tab pane shows the following names: ◦ For Teradata: ▪ The database name is the name that you enter in the `collibraSystemName` property. ▪ The schema name is the name that you enter in the `databaseNames` property. ◦ For MySQL: ▪ The database name is the name that you enter in the `databaseNames` property.

Properties	Description
connectAsServiceName	The option to determine whether your Oracle database uses an Oracle service name or SID. ◦ `True`: Connect to an Oracle database that uses an Oracle service name. Enter the service name in the `databaseNames` property. ◦ `False`: Connect to an Oracle database that uses an SID. Enter the SID in the `databaseNames` property. > Note This property is only valid for Oracle databases. It will be ignored for all other databases.
hostname	The name of your database host.

Properties	Description
collibraSystemName	The name of the data source's system or server. This is also the full name of your System asset in Data Catalog. You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog. If the `useCollibraSystemName` property is: ◦ `false` (default), system or server names in table references in analyzed SQL code are ignored. This means that a table that exists in two different systems or servers is identified (either correctly or incorrectly) as a single data object, with a single asset full name. ◦ `true`, system or server names in table references are considered to be represented by different System assets in Data Catalog. The value of the `collibraSystemName` field is used as the default system or server name.
port	The port number.

Properties	Description
customConnectionProperties	An option to enable the lineage harvester to read additional connection parameters. This parameter is only required in very specific situations. If you don't need it, you can remove it from the configuration file. Note You can currently only use this property for the following data sources: ◦ HiveQL ◦ IBM DB2 ◦ Netezza ◦ PostgreSQL ◦ Redshift ◦ SAP Hana ◦ Snowflake ◦ Spark SQL ◦ Sybase
<Google BigQuery database>	This configuration section contains the required information for a Google BigQuery database.
id	The unique ID of your data source. For example, `my_third_data_source`.
type	The kind of data source. In this case, the value has to be `DatabaseBigQuery`.
projectIDs	The IDs of your Google BigQuery project. You can add multiple projects. For example, `[ "first-project", "second-project", "third-project" ]`. Note You have to use the same project ID as the full name of the Database asset that you create when you prepare the physical data layer in Data Catalog.

Properties	Description
region	The location of your BigQuery data. This is the region that you specified when you create a data set. You can only add one location as value. However, you can create separate BigQuery entries per location in the configuration file. As a result, you create a complete technical lineage with Google BigQuery data from different locations. Note This property is optional.
auth	The path to a JSON file that contains authentication information. Tip For more information about setting up the authentication, see the Google Big Query user guide.
collibraSystemName	The name of the Google BigQuery system. This is also the full name of your System asset in Data Catalog. You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog.
<Snowflake database>	This configuration section contains the required information for a Snowflake database.
id	The unique ID of your data source. For example, `my_fourth_data_source`.

Properties	Description
type	The kind of data source. In this case, the value has to be `DatabaseSnowflake`.
username	The username that you use to sign in to your data source.
hostname	The URL that you use to access Snowflake web console. For example, `<AccountName>.snowflakecomputing.com`.
collibraSystemName	The name of the Snowflake system. This is also the full name of your System asset in Data Catalog. You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog.
databaseNames	The names of your databases. Enter the database names of your data source between double quotes ("") and put everything between square brackets. If you want to include more than one database, separate them by a comma. For example, ["`MyFirstSnowflakeDatabase`", "`MySecondSnowflakeDatabase`"] Note You have to use the same database names as the full names of the Database assets that you create when you prepare the physical data layer in Data Catalog.

Properties	Description
warehouse	The name of your virtual warehouse. Note This property is optional.
customConnectionProperties	An option to enable the lineage harvester to read additional connection parameters. This parameter is only required in very specific situations. If you don't need it, you can remove it from the configuration file. Example If you get an OSCP scan error, you can turn OSCP checking off by using the following value: `insecureMode=true`.
<SQL files in the lineage harvester output folder>	This configuration section contains the required information for SQL files of a data source that were previously downloaded by the lineage harvester and is stored in the lineage harvester output folder.
type	The kind of data source. In this case, the value has to be `LoadedSource`.
id	The unique ID of the data source that you uploaded to the lineage harvester folder. For example, `my_loaded_snowflake_source`.
zipFile	The full path to the ZIP file that was created in the lineage harvester folder.
<Tableau>	This configuration section contains the required information for Tableau integration.
sources	This section contains all Tableau connection properties.

Properties	Description
type	The kind of data source. In this case, the value has to be *Tableau*.
id	The unique ID to identify the Tableau metadata that was uploaded to the Collibra Data Lineage. Tip This value can be anything as long as it is a unique. The lineage harvester uses the ID to identify a batch of data on the Collibra Data Lineage server.
url	The link to the data in Tableau.
username	The username you use to sign in to the Tableau server. Important If you want to use token-based authentication, you need to replace `username` with `tokenName`. You must specify either `username` or `tokenName`; if both exist, then `tokenName` is used.
tokenName	The lineage harvester authentication token. Note For token-based authentication, use this property in your lineage harvester configuration file, instead of the `username` property. If both properties are present, `tokenName` is used.

Properties	Description
siteIds	The site IDs of the Tableau sites that you want to include in the ingestion process. Warning Ensure that you specify the correct value. The correct value is the URL of the site to which you want to sign in. When you manually sign in to Tableau Server or Tableau Online, the site ID is the value that appears after **/site/** in the browser address bar. In the following example URLs, the site ID is `MarketingTeam`: ◦ Tableau Server: http://MyServer/#/site/MarketingTeam/projects ◦ Tableau Online: https://10ay.online.tableau.com/#/site/MarketingTeam/workbooks Example If you want to ingest two Tableau sites "Site 1" and "Site 2", you can enter the following information in the siteIds property: ["site ID of Site 1", "site ID of Site 2"].
siteNames	The site names of the corresponding site IDs. Important This property is: ◦ Optional for Tableau Server ◦ Mandatory for Tableau Online.

Properties	Description
restOnly	Indication whether or not you would like to use both the Tableau REST API and Tableau Metadata API to harvest Tableau metadata. ◦ `false` (default): The lineage harvester will use the REST API and Metadata API to harvest Tableau metadata. ◦ `true`: The lineage harvester will only use the REST API to harvest Tableau metadata. Warning If you only allow the lineage harvester to use the Tableau REST API, the harvester won't be able to process the necessary information for the technical lineage and the automatic stitching of Column assets to Tableau Data Attribute assets will not be possible.
collibraSystemName	The name of the data source's system or server. If the `useCollibraSystemName` property is set to `true`, you must prepare a configuration file to provide the system information.

Properties	Description
domainId	The unique ID of the domain in Collibra Data Intelligence Cloud in which you want to ingest the Tableau assets. Finding the domain ID a. Open the domain. b. Copy the domain ID. Tip If you go to your domain, you can find the domain ID in the URL. The URL looks like: https://<yourcollibrainstance>/domain/**22258f64-40b6-4b16-9c08-c95f8ec0da26**?view=00000000-0000-0000-0000-000000040001. In this example, the domain ID is in bold.
excludeImages	Optional parameter for excluding the downloading of images. To exclude the downloading of images, set this property to `true`.

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td>paging</td><td>
Optional parameter for customizing the Tableau API pagination settings.
The default values are sufficient in most cases; however, you can decrease them to help mitigate node limit errors, or increase them to speed up API calls.

The complete list of pagination settings, descriptions and default values
<pre>
"paging": {
   "databasesPageSize": 100,
   "tablesPageSize": 100,
   "tablesColumnsPageSize": 100,
   "tableColumnsPageSize": 1000,
   "datasourcesPageSize": 50,
   "datasourcesFieldsPageSize": 50,
   "datasourceFieldsPageSize": 100,
   "worksheetsPageSize": 100,
   "worksheetsFieldsPageSize": 100,
   "worksheetFieldsPageSize": 1000,
   "dashboardsPageSize": 100,
   "columnsLimit": 20,
   "fieldsLimit": 20
   }
</pre>
Settings per metadata type and descriptions
<table>
<tr><th>Metadata type</th><th>Setting and description</th></tr>
<tr><td>Dashboard</td><td>◦ <code>dashboardsPageSize</code>: The number of dashboards per page.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Metadata type</th><th>Setting and description</th></tr>
<tr><td>Worksheet</td><td>◦ <code>worksheetsPageSize</code>: The number of worksheets per page.
◦ <code>worksheetsFieldsPageSize</code>: The number of worksheet fields per page.</td></tr>
<tr><td>Database</td><td>◦ <code>databasesPageSize</code>: The number of databases per page.</td></tr>
<tr><td>Table</td><td>◦ <code>tablesPageSize</code>: The number of tables per page.
◦ <code>tablesColumnsPageSize</code>: The number of table columns per page.</td></tr>
<tr><td>Table columns</td><td>◦ <code>tableColumnsPageSize</code>: The number of table columns per page.</td></tr>
<tr><td>Data source</td><td>◦ <code>datasourcesPageSize</code>: The number of data sources per page.
◦ <code>datasourcesFieldsPageSize</code>: The number of data source fields per page.
◦ <code>columnsLimit</code>: The number of data source field columns per page.
◦ <code>fieldsLimit</code> : The number of referenced data source fields per page.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Metadata type</th><th>Setting and description</th></tr>
<tr><td>Data source field</td><td>

<code>datasourceFieldsPageSize</code>: The number of data source fields per page.
<code>columnsLimit</code>: The number of data source field columns per page.
<code>fieldsLimit</code> : The number of referenced data source fields per page.

</td></tr>
</table>
</td></tr>
<tr><td><Power BI></td><td>This configuration section contains the required information for Power BI integration.

Note You have to purchase the Power BI connector and lineage feature. Then you need to add the Power BI connection properties to both the lineage harvester configuration file and the Power BI harvester configuration file to ingest Power BI metadata into Data Catalog.</td></tr>
<tr><td>type</td><td>The kind of data source. In this case, the value has to be <code>ExistingLineage</code>.</td></tr>
<tr><td>id</td><td>The unique ID of the Power BI metadata you harvested via the Power BI harvester.

You must use the same ID as the value you used in the Power BI configuration file sourceID property.</td></tr>
<tr><td><Looker></td><td>This configuration section contains the required information for Looker integration.</td></tr>
</table>

Properties	Description
collibraSystemName	The name of the Looker system or server. If the `useCollibraSystemName` property is set to `true`, you must prepare a configuration file to provide the system information.
id	The unique ID of your Looker metadata. For example, *my_looker*. Tip This value can be anything as long as it is unique and human readable. The ID identifies the batch of Looker metadata on the Collibra Data Lineage server.
type	The kind of data source. In this case, the value has to be *Looker*.
lookerUrl	The URL to your Looker API. Tip There are two ways to find the Looker API URL: ◦ In the API Host URL field in the Looker Admin menu. If this field is empty, you can use the default Looker API URL which you can find in the interactive API documentation. ◦ In the interactive API documentation URL. It is the part of the URL before `/api-docs/`.
clientId	The username you use to access the Looker API.
domainId	The unique ID of the domain in Collibra Data Intelligence Cloud in which you want to ingest the Looker assets.

3. Save the configuration file.

4. Start the lineage harvester again and do one of the following:
   - To process data from all data sources in the configuration file, run the following command:
     For windows:

     ```
     .\bin\lineage-harvester.bat full-sync
     ```

     For other operating systems:

     ```
     ./bin/lineage-harvester full-sync
     ```

   - To process data from specific data sources in the configuration file, run the following command:
     For windows:

     ```
     .\bin\lineage-harvester.bat full-sync -s "ID of the data source"
     ```

     For other operating systems:

     ```
     ./bin/lineage-harvester full-sync -s "ID of the data source"
     ```

     » The lineage harvester sends the data source information to a Collibra Data Lineage server using Collibra REST API, where it is parsed and analyzed. As a result, the technical lineage is created and shown in Data Catalog.
5. When prompted, enter the passwords to connect to Collibra and your data sources.
   Do one of the following:
   - Enter the passwords in the console.
     » The passwords are encrypted and stored in **/config/pwd.conf**.
   - Provide the passwords via command line.
     » The passwords are stored locally and not in your lineage harvester folder.

> Tip If the lineage harvester log shows an error message or the harvesting process fails, you can use the technical lineage troubleshooting guide to fix your issue.

What's next?

If you prepared the physical data layer and have the required permissions, you can go to the asset page of a Table, Column Power BI Column or Looker Look asset from the data source that you added in the configuration file and visualize the technical lineage. The

technical lineage shows the data source information of data sources that have been successfully analyzed and processed.

The lineage harvester can also use scheduled jobs to synchronize the data sources on fixed times.

> Tip You can check the progress of the technical lineage creation in Activities. The **Results** field indicates how many relations were imported into Data Catalog. Go to the status page to see the log files of the SQL analysis.

The configuration file generator

The configuration file generator helps you create your lineage harvester configuration file more easily by providing the structure of the file with the correct properties per data source.

The lineage harvester configuration file

The lineage harvester uses a configuration file when it connects to Data Catalog via Collibra REST API. The configuration file contains references to the data sources for which you want to create a technical lineage. You have to prepare the configuration file if you want to create a technical lineage and add new relations of the type "Data Element targets / sources Data Element" between existing assets in Data Catalog and "Column is target of / is source of Data Attribute" between assets from ingested BI sources and assets in Data Catalog.

> Tip You have to save the configuration file in the **config** directory in the lineage harvester folder.

Empty configuration file

When you run the lineage harvester for the first time, it creates an empty configuration file. To create a technical lineage, you have to manually add properties and values, per data source, to this configuration file.

The following image shows an example of the empty configuration file created by the lineage harvester.

```
{
   "general" : {
       "catalog" : {
           "url" : "",
           "username" : "",
       },
       "useCollibraSystemName" : false
   },
   "sources" : [ {
       "type" : "Database",
       "id" : "MyDB",
       "hostname" : "",
       "username" : "",
       "dialect" : "",
       "collibraSystemName" : "",
       "databaseNames" : [ ],
       "port" : 1521
   } ]
}
```

Configuration file generator

> Tip The configuration file generator is only available in the online version of the user guide.

The configuration file generator creates an example configuration file with the data source properties of your choosing:

1. Scroll down to the configuration file example.
2. Paste the example in your empty configuration file in the lineage harvester**config** folder.
3. Replace the values in the example to match your actual data source information.

   > Tip Make sure you understand each property and know which values you must use to access your data source information.

4. Run the lineage harvester.

> Warning Some browser plug-ins may slow the configuration file generator down.

```
{
   "general": {
       "catalog" : {
           "url" : "https://companydomain.collibra.com",
           "username" : "my-Collibra-username"
           },
           "useCollibraSystemName" : false
   },
   "sources" : [

   {
       "collibraSystemName" : "datastage-system-name",
       "id" : "datastage_source",
       "type" : "ExternalDirectory",
       "dirType" : "DATASTAGE",
       "path" : "/path/to/the/datastage/folder/",
       "mask" : "*",
       "recursive" : false
   }
   {
       "collibraSystemName" : "infa-system-name",
       "id" : "informatica_source",
       "type" : "ExternalDirectory",
       "dirType" : "INFA",
       "path" : "/path/to/the/informatica/folder/",
       "mask" : "*",
       "recursive" : false
   }
   {
       "collibraSystemName" : "ssis-system-name",
       "id" : "datastage_source",
       "type" : "ExternalDirectory",
       "dirType" : "SSIS",
       "path" : "/path/to/the/ssis/folder/",
       "mask" : "*",
       "recursive" : false
   }
   {
       "type" : "IICS",
       "id" : "iics_source",
       "collibraSystemName" : "iics-development",
       "loginUrl" : "https://dm-us.informaticaintelligentcloud.com",
       "username" : "login-iics"
       "objects" : [
           {
               "path" : "Default/Sales",
               "type" : "Project"
           },
           {
               "path" : "My Project/Statistics",
```

```
                "type" : "Project"
            }
        ]
    }
    {
        "id" : "my-matillion-project",
        "type" : "Matillion",
        "url" : "https://my-domain",
        "groupName" : "my-matillion-group",
        "projectName" : "redshift-project",
        "environmentName" : "redshift-environment",
        "dialect" : "redshift",
        "username" : "ec2-user",
        "startTimestamp" : 1594080796911,
        "collibraSystemName": "Matillion-system"
    }
    {
        "type": "Tableau",
        "id": "unique-ID",
        "url": "URL to Tableau server?",
        "username": "Admin",
        "siteIds": ["site ID of Tableau Site 1", "site ID of Tableau Site
2"],
        "siteNames": ["site name of Tableau Site 1", "site name of Tableau
Site 2"],
        "restOnly": false,
        "collibraSystemName": "tableau-system-name",
        "domainId": "Domain-resource-ID",
        "excludeImages": true,
        "paging": {
            "pagination-setting": 100,
            "pagination-setting-2": 100
        }
    }
    {
        "collibraSystemName" : "looker",
        "id" : "looker-source",
        "type" : "Looker",
        "lookerUrl" : "https://<instance-name.api.looker.com",
        "clientId" : "my-looker-api-user-name",
        "domainId" : "22258f64-40b6-4b16-9c08-c95f8ec0da26"
    }
    {
        "type" : "ExistingLineage",
        "id" : "MyPowerBISourceID"
    }
    {
        "collibraSystemName" : "custom-system-name",
        "id" : "MyCustomLineage",
        "type" : "ExternalDirectory",
```

```
        "dirType" : "custom-lineage",
        "path" : "/path/to/custom-lineage/dir/file.json"
    }
    {
        "type" : "LoadedSource",
        "id" : "MySource",
        "zipFile" : "/path/to/source-MySource.zip"
    }
    {
        "id" : "database_source",
        "type" : "Database",
        "username" : "MyUsername",
        "dialect" : "hive",
        "databaseNames" : ["MyDefaultDbName"],
        "hostname" : "localhost",
        "collibraSystemName" : "apache-hive-system",
        "port" : 1521,
        "customConnectionProperties" : ""
    }
    {
        "id" : "oracle_source",
        "type" : "Database",
        "username" : "MyUsername",
        "dialect" : "oracle",
        "databaseNames" : ["oracle-service-name"],
        "connectAsServiceName" : true,
        "hostname" : "localhost",
        "collibraSystemName" : "oracle-system-name",
        "port" : 1521
    }
    {
        "id" : "bigquery_source",
        "type" : "DatabaseBigQuery",
        "projectIDs" : [ "bigquery_project1", "bigquery_project2" ],
        "region": "europe-west1"
        "auth" : "/path/to/the/authentication/file.json",
        "collibraSystemName" : "bigquery-system-name"
    }
    {
        "id" : "snowflake_source",
        "type" : "DatabaseSnowflake",
        "username" : "MyUsername",
        "hostname" : "MyAccountName.snowflakecomputing.com",
        "collibraSystemName" : "snowflake-system-name",
        "databaseNames" : ["MyFirstDbName","MySecondDbName"],
        "warehouse" : "MySnowflakeWarehouseName",
        "customConnectionProperties" : ""
    }
    {
        "id" : "sqldirectory_source",
```

```
        "type" : "SqlDirectory",
        "path" : "/path/to/the/sql/folder/",
        "mask" : "*",
        "recursive" : false,
        "dialect" : "db2",
        "database" : "MyDefaultDbName",
        "collibraSystemName" : "data-source-system",
        "schema" : "MyDefaultDbSchema",
        "verbose" : true
    } ]
}
```

> Important If you want to ingest Power BI in Data Catalog you need both the Power BI harvester and the lineage harvester. You can find more information about the Power BI harvester configuration file and Power BI source ID configuration file in the Power BI section of the documentation.

Informatica PowerCenter

The following example shows an Informatica PowerCenter <source ID> configuration file.

```
{
    "connectionDefinitions": {
        "oracle_source": {
            "dbname": "oracle-source-database-name1",
            "schema": "my Oracle source schema",
            "dialect": "oracle"
        },
        "oracle_target": {
            "dbname": "oracle-target-database-name2",
            "schema": "my other oracle target schema",
            "dialect": "oracle"
        }
    },
    "collibraSystemNames": {
        "databases": [
            {
                "dbname": "oracle-source-database-name1",
                "collibraSystemName": "oracle-system-name1"
            },
            {
                "dbname": "oracle-target-database-name2",
                "collibraSystemName": "oracle-system-name2"
            }
        ],
        "connections": [
```

```
            {
                "connectionName": "oracle-connection-name1",
                "collibraSystemName": "oracle-system-name1"
            },
            {
                "connectionName": "oracle-connection-name2",
                "collibraSystemName": "oracle-system-name2"
            }
        ]
    }
}
```

SQL Server Integration Services

The following example shows an SQL Server Integration Services connection definitions configuration file.

```
{
  "ConnStringRegExTranslation": {

    "Data Source=dhb-sql-prod;Initial Catalog=SFG_repl_
staging;Provider=SQLNCLI11;Integrated Security=SSPI.*": {
      "dbname": "DATAHUB",
      "schema": "DBO",
      "dialect": "mssql",
      "collibraSystemName" : "WAREHOUSE"
    },

    "Server=sb-dhub;User ID=SYS_USER;Initial
Catalog=STAGEDB;Port=6306.*": {
      "dbname": "STAGEDB",
      "schema": "STAGE_OWNER",
      "dialect": "sybase",
      "collibraSystemName" : ""
    }

  }
}
```

IBM InfoSphere DataStage

The following example shows a DataStage connection definitions configuration file.

```
{
  "OdbcDataSources": {
    "oracle-data-source": {
```

```
      "dbname": "my-oracle-database",
      "schema": "my-oracle-schema",
      "dialect": "oracle",
      "collibraSystemName": "my-system"
    },
    "mssql-data-source": {
      "dbname": "my-mssql-database",
      "schema": "my-mssql-schema",
      "dialect": "mssql",
      "collibraSystemName": "my-system"
    }
  },
  "NonOdbcConnectors": {

    "admin@database-name": {
      "dbname": "my-netezza-database",
      "schema": "my-netezza-schema",
      "dialect": "netezza",
      "collibraSystemName": "my-system"
    },
    "admin@second-database-name": {
      "dbname": "my-second-netezza-database",
      "schema": "my-second-netezza-schema",
      "dialect": "netezza",
      "collibraSystemName": "my-system"
    }
  }
}
```

Informatica Intelligent Cloud Services

The following example shows an Informatica Intelligent Cloud Services <source ID> configuration file.

```
{
   "collibraSystemNames": {
       "connections": [
           {
               "connectionName": "DG_con_standby_cmdm_clientors",
               "collibraSystemName": "PUBLIC"
           },
           {
               "connectionName": "DG_con_dev_dg_dgiauser_su",
               "collibraSystemName": "PUBLIC"
           }
       ]
   },
   "connectionDefinitions": [
```

```
        {
            "connectionName": "DG_con_standby_cmdm_clientors",
            "databaseName": "main",
            "schemaName": "dbo",
            "dialect": "oracle"
        },
        {
            "connectionName": "DG_con_dev_dg_dgiauser_su",
            "databaseName": "main",
            "schemaName": "dbo",
            "dialect": "oracle"
        }
    ]
}
```

Tableau

The following example shows a Tableau <source ID> configuration file.

```
{
  "collibraSystemNames": {
    "databases": [
      {
        "hostName": "tableau-server.us-east-1.rds.amazonaws.com",
        "collibraSystemName": "public"
      }
    ],
    "files": [
      {
        "filePath": "C:\ProgramData\Tableau\Tableau
Server\data\files\sample.xls",
        "collibraSystemName": "sample-files"
      }
    ],
    "connectors": [
      {
        "connectorUrl": "tableau-server-connector-url.com",
        "collibraSystemName": "Oracle-connector"
      }
    ],
    "cloudFiles": [
      {
        "name": "file-name",
        "collibraSystemName": "FILE"
      }
    ]
  },
  "catalogRelationsIds": {
    "Tableau Workbook contains/contained in Tableau Data Model":
```

```
"12345678-abc1-def2-4ef5-abcdef012345",
    "Report uses/used in Data Attribute": "0987654-abc2-def3-4ef5-
abcdef67890"
  }
}
```

Looker

The following example shows a Looker <source ID> configuration file.

```
{
   "Connections": {
       "connection-object1": {
           "dialect": "mssql",
           "schema": "mssql-schema-name",
           "dbname": "mssql-database-name",
           "collibraSystemName": "mssql-system-name"
       },
       "connection-object2": {
           "dialect": "oracle",
           "schema": "oracle-schema-name",
           "dbname": "oracle-database-name",
           "collibraSystemName": "oracle-system-name"
       }
   }
}
```

Prepare an external directory folder

If you want to create a technical lineage for an external directory such as Informatica PowerCenter, SQL Server Integration Services (SSIS) or IBM InfoSphere DataStage, you must prepare a folder with the external directory's data source files.

If the external directory files do not have the necessary information, for example a database and a schema, to stitch the data sources, you have to provide the connection definitions manually via a JSON configuration file. This is required at each connection, regardless of whether the `useCollibraSystemName` property in the lineage harvester configuration file is set to `true` or `false`.

> Tip Go to the online version of the user guide for more detailed steps and examples.

> Note You can also create and configure a JSON file to define a custom technical lineage.

Prerequisites

- You have IBM InfoSphere Information Server version 11.5 or newer.
- You have Informatica PowerCenter version 9.6 or newer.
- You have SQL Server Integration Services 2012 or newer with package format version 6 or newer.
- You have Microsoft Visual Studio version 2012 or newer.
- You have downloaded the lineage harvester and you have the necessary system requirements to run it.
- You have prepared the physical data layer in Data Catalog.

  > Note To stitch the data objects in the source and target data sources in external directories with Data Catalog assets, you first have to register those data sources in Data Catalog.

> Tip If you want to create a technical lineage for Informatica Intelligent Cloud Services Data Integration, you don't have to create a folder with data source files. You add your data source information directly to the lineage harvester configuration file.

Steps to create a technical lineage for Informatica PowerCenter

1. Create a local folder.
2. Export the Informatica objects or repository for which you want to create a technical lineage to the local folder.

Note
- All XML and parameter files, for example PAR, TXT or PRM files in this folder and its subfolders are taken into account when you create a technical lineage, but Collibra Data Lineage only shows a technical lineage for workflows that have mappings with sources, transformations and targets. Collibra supports the most common Informatica PowerCenter transformations. For more information, see the Informatica PowerCenter documentation.
- A technical lineage is created when the following tags are present in your XML file:
  - <REPOSITORY>
  - <FOLDER>
  - <SOURCE> / <TARGET>
  - <SESSION>
  - <MAPPING>
  - <TRANSFORMATION> (within a <MAPPING> tag)

3. Put your parameter files in the right location.

If...	Then...
all parameter files are PAR files	No action required
not all parameter files are PAR files	a. Create a new folder in the local folder. b. Name the folder *techlin-param*. c. Move all parameter files that are used by the exported XML to the **techlin-param** folder. d. In the lineage harvester configuration file, set the `recursive` property to `true`. Note The lineage harvester only takes into account parameter files in the **techlin-param** folder.

4. Optionally, create a source ID configuration file with connection definitions and system names:

> Tip If you previously created a technical lineage for Informatica PowerCenter with connection definitions, the connection_definitions.conf file will still be taken into account.

a. Create a new JSON file in the lineage harvester **config** folder.
b. Give the JSON file the same name as the value of the `Id` property in the lineage harvester configuration file.

> Example The value of the `Id` property in the lineage harvester configuration file is `informatica-source-1`. As a result, the name of your JSON file should be *informatica-source-1.conf*.

c. For each data source, add the following content to the JSON file:

Property	Description
connectionDefinitions	This section contains the connection properties to a source in Informatica PowerCenter.
<connectionName>	The type of your source or target data source. This section contains the connection properties to a source or target in Informatica PowerCenter.
dbname	The name of your source or target database.
schema	The name of your source or target schema.

Property	Description
dialect	The dialect of the referenced database. Tip You can enter one of the following values: ■ *azure*, for an Azure SQL Server data source. ■ *bigquery*, for a Google BigQuery data source. ■ *db2*, for an IBM DB2 data source. ■ *hana*, for a SAP Hana data source. ■ *hana-cviews*, for SAP Hana data calculation views. ■ *hive*, for a HiveQL data source. ■ *greenplum*, for a Greenplum data source. ■ *mssql*, for a Microsoft SQL Server data source. ■ *mysql*, for a MySQL data source. ■ *netezza*, for a Netezza data source. ■ *oracle*, for an Oracle data source. ■ *postgres*, for a PostgreSQL data source. ■ *redshift*, for an Amazon Redshift data source. ■ *snowflake*, for a Snowflake data source. ■ *spark*, for a Spark SQL data source. ■ *sybase*, for a Sybase data source. ■ *teradata*, for a Teradata data source.

Property	Description
collibraSystemNames	This section contains the system or server name that is specified in your database and referenced in your connection. Note This section is only required when the useCollibraSystemName flag in the lineage harvester configuration file is set to `true`.
databases	This section contains the database information. This is required to connect directly to the system or server of the database.
dbname	The name of the database. The database name is the same as the name you entered in the <connectionName> section.
collibraSystemName	The system or server name of the database.
connections	This section contains the connection information. This is required to reference to the system or server of the connection.
connectionName	The name of the connection.
collibraSystemName	The system or server name of the connection.

Important If you are using variables in Informatica PowerCenter, add the value of the variable instead of the name in the connection definitions JSON file. For example, if the parameter file contains `$DBConnection_`

`dwh=DWH_EXPORT` then you add the following connection definitions to the JSON file:

```
{
   "DWH_EXPORT":

       { "dbname": "DWH", "schema": "DBO" }
}
```

5. Add a new section for Informatica PowerCenter to the lineage harvester configuration file.

Example of the connection_definitions.conf file

```
{
   "connectionDefinitions": {
       "oracle_source": {
           "dbname": "oracle-source-database-name1",
           "schema": "my Oracle source schema",
           "dialect": "oracle"
       },
       "oracle_target": {
           "dbname": "oracle-target-database-name2",
           "schema": "my other oracle target schema",
           "dialect": "oracle"
       }
   },
   "collibraSystemNames": {
       "databases": [
           {
               "dbname": "oracle-source-database-name1",
               "collibraSystemName": "oracle-system-name1"
           },
           {
               "dbname": "oracle-target-database-name2",
               "collibraSystemName": "oracle-system-name2"
           }
       ],
       "connections": [
           {
               "connectionName": "oracle-connection-name1",
               "collibraSystemName": "oracle-system-name1"
           },
           {
               "connectionName": "oracle-connection-name2",
               "collibraSystemName": "oracle-system-name2"
           }
```

```
        ]
    }
}
```

Steps to create a technical lineage for SQL Server Integration Services

1. Create a local folder.
2. Export the SSIS files for which you want to create a technical lineage.

   > Tip You can export them directly from the SQL Server Integration Services repository or via Microsoft Visual Studio. For more information, see the SQL Server Integration Services documentation.

3. Store the SSIS files to your local folder. Typically, the folder contains the following files:
   - SSIS package files (DTSX), containing the SQL Server Integration Services source code.
   - Connection manager files (CONMGR), containing environment and connection information.
   - Parameter files (PARAMS), if applicable.

   > Note All files in this folder and subfolders are taken into account when you create a technical lineage. The lineage harvester automatically detects data sources in the SSIS files.

4. Optionally, configure the connection definitions:

   > Tip If the `useCollibraSystemName` in the lineage harvester configuration file is set to `true`, you must provide the connection_definitions.conf file.

   a. Create a new JSON file in the local folder.
   b. Name the JSON file *connection_definitions.conf*.

c. For each supported data source, specify the relevant translations.

Property	Description
ConnStringRegExTranslation	The parent element that opens the connection definitions.

Property	Description
<regular expression>	A regular expression that must match one or more connection strings. Note Important considerations: ■ By default, the regular expression is not case sensitive. As a consequence, a regular expression can match with connection strings containing uppercase characters or lowercase characters. ■ The connection string is part of the SSIS connection manager. ■ SSIS connection managers are included in an SSIS package files (DTSX) or in connection manager files (CONMGR).

Property	Description
	Example Regular expression: `Server=sb-dhub;User ID=SYB_USER2;Initial Catalog=STAGEDB;Port=6306.*` Explanation: The first section, up to .*, is a literal, but not case-sensitive, match of the characters. The dot (.) can match any single character. The asterisk (*) means zero or more of the previous, in this case any character. Match: Any connection string that starts with `Server=sb-dhub;User ID=SYB_USER2;Initial Catalog=STAGEDB;Port=6306.` Example: `Server=sb-dhub;User ID=SYB_USER2;Initial Catalog=STAGEDB;Port=6306;Persist Security Info=True;Auto Translate=False;.`
dbname	The name of your database, to which the data source connection refers.
schema	The name of your schema, to which the regular expression refers.

Property	Description
dialect	The dialect of the referenced database. Tip You can enter one of the following values: ■ *azure*, for an Azure SQL Server data source. ■ *bigquery*, for a Google BigQuery data source. ■ *db2*, for an IBM DB2 data source. ■ *hana*, for a SAP Hana data source. ■ *hana-cviews*, for SAP Hana data calculation views. ■ *hive*, for a HiveQL data source. ■ *greenplum*, for a Greenplum data source. ■ *mssql*, for a Microsoft SQL Server data source. ■ *mysql*, for a MySQL data source. ■ *netezza*, for a Netezza data source. ■ *oracle*, for an Oracle data source. ■ *postgres*, for a PostgreSQL data source. ■ *redshift*, for an Amazon Redshift data source. ■ *snowflake*, for a Snowflake data source. ■ *spark*, for a Spark SQL data source. ■ *sybase*, for a Sybase data source. ■ *teradata*, for a Teradata data source.

Property	Description
collibraSystemName	The name of the referenced data source's system or server. This property is only required when you set the useCollibraSystemName property in the lineage harvester configuration file to `true`. If this property is set to `false`, you can remove the collibraSystemName property or enter an empty string. > Note You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog. If the “useCollibraSystemName" property is: ■ `false`, system or server names in table references in analyzed SQL code are now ignored. This means that a table that exists in two different systems or servers is identified (either correctly or incorrectly) as a single data object, with a single asset full name. ■ `true`, system or server names in table references are considered to be represented by different System assets in Data Catalog. The value of the "collibraSystemName" field is used as the default system or server name.

5. Add a section for SQL Server Integration Services to the lineage harvester configuration file.

Example of the connection_definitions.conf file

```
{
  "ConnStringRegExTranslation": {

    "Data Source=dhb-sql-prod;Initial Catalog=SFG_repl_
staging;Provider=SQLNCLI11;Integrated Security=SSPI.*": {
      "dbname": "DATAHUB",
      "schema": "DBO",
      "dialect": "mssql",
      "collibraSystemName" : "WAREHOUSE"
    },

    "Server=sb-dhub;User ID=SYS_USER;Initial
Catalog=STAGEDB;Port=6306.*": {
      "dbname": "STAGEDB",
      "schema": "STAGE_OWNER",
      "dialect": "sybase",
      "collibraSystemName" : ""
    }

  }
}
```

Steps to create a technical lineage for DataStage

1. Create a local folder.
2. Export the DataStage project files (DSX) for which you want to create a technical lineage.

   > Tip You can either export a DataStage project manually or automatically via command line.

3. Store the DataStage files in your local folder.
4. Optionally, if your DataStage project uses environment variables, manually export the environment files (ENV).
5. Give the environment files the same name as the DataStage project files. For example, if your project file is named *datastage-project-1.dmx*, you have you name your environment file *datastage-project-1.env*.

6. Store the environment files in the same local folder.

> Important
> - The lineage harvester only supports DSX and ENV files.
> - You can have one DSX file per DataStage project.
> - You can have one or none ENV file per DSX file.
> - The name of the DSX file and the ENV file has to be the same.

7. Optionally, configure the connection definitions:
   a. Create a new JSON file in the local folder.
   b. Name the JSON file *connection_definitions.conf*.
   c. For each data source, specify the relevant translations:

Property	Description
OdbcDataSources	Open Database Connectivity data sources in IBM InfoSphere DataStage for which you want to create a technical lineage.
<data-source-name>	The ODBC data source name that you use in your DataStage projects. This section contains the properties to translate the database, schema and dialect.
dbname	The name of your database, to which the ODBC data source connection refers.
schema	The name of your schema, to which the ODBC data source connection refers.

Property	Description
dialect	The dialect of the referenced database. Tip You can enter one of the following values: ■ *azure*, for an Azure SQL Server data source. ■ *bigquery*, for a Google BigQuery data source. ■ *db2*, for an IBM DB2 data source. ■ *hana*, for a SAP Hana data source. ■ *hana-cviews*, for SAP Hana data calculation views. ■ *hive*, for a HiveQL data source. ■ *greenplum*, for a Greenplum data source. ■ *mssql*, for a Microsoft SQL Server data source. ■ *mysql*, for a MySQL data source. ■ *netezza*, for a Netezza data source. ■ *oracle*, for an Oracle data source. ■ *postgres*, for a PostgreSQL data source. ■ *redshift*, for an Amazon Redshift data source. ■ *snowflake*, for a Snowflake data source. ■ *spark*, for a Spark SQL data source. ■ *sybase*, for a Sybase data source. ■ *teradata*, for a Teradata data source.

Property	Description
collibraSystemName	The name of the data source's system or server. This property is only required when you set the useCollibraSystemName property in the lineage harvester configuration file to `true`. If this property is set to `false`, you can remove the collibraSystemName property or enter an empty string. > Note You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog.
NonOdbcConnectors	Other data source connectors in IBM InfoSphere DataStage for which you want to create a technical lineage. For example, DB2, Oracle or Netezza. > Note This section is optional.

Property	Description
<data-source-connector-ID>	The data source username and database of the connector that you use in your DataStage projects. This usually looks like for example *admin@database-name*. The combination of the username and database name should be unique. The following section contains the properties to translate the database, schema and dialect.
dbname	The name of your database, to which the data source connection refers.
schema	The name of your schema, to which the data source connection refers.

Property	Description
dialect	The dialect of the referenced database. Tip You can enter one of the following values: ■ *azure*, for an Azure SQL Server data source. ■ *bigquery*, for a Google BigQuery data source. ■ *db2*, for an IBM DB2 data source. ■ *hana*, for a SAP Hana data source. ■ *hana-cviews*, for SAP Hana data calculation views. ■ *hive*, for a HiveQL data source. ■ *greenplum*, for a Greenplum data source. ■ *mssql*, for a Microsoft SQL Server data source. ■ *mysql*, for a MySQL data source. ■ *netezza*, for a Netezza data source. ■ *oracle*, for an Oracle data source. ■ *postgres*, for a PostgreSQL data source. ■ *redshift*, for an Amazon Redshift data source. ■ *snowflake*, for a Snowflake data source. ■ *spark*, for a Spark SQL data source. ■ *sybase*, for a Sybase data source. ■ *teradata*, for a Teradata data source.

Property	Description
collibraSystemName	The name of the data source's system or server. This property is only required when you set the useCollibraSystemName property in the lineage harvester configuration file to `true`. If this property is set to `false`, you can remove the collibraSystemName property or enter an empty string. You must use the same system name as the full name of the System asset that you create when you prepare the physical data layer in Data Catalog. If you don't prepare the physical data layer, Collibra Data Lineage cannot stitch the data objects in your technical lineage to the assets in Data Catalog.

8. Add a section for IBM InfoSphere DataStage to the lineage harvester configuration file.

Example of the connection_definitions.conf file

```
{
  "OdbcDataSources": {
    "oracle-data-source": {
      "dbname": "my-oracle-database",
      "schema": "my-oracle-schema",
      "dialect": "oracle",
      "collibraSystemName": "my-system"
    },
    "mssql-data-source": {
      "dbname": "my-mssql-database",
      "schema": "my-mssql-schema",
      "dialect": "mssql",
      "collibraSystemName": "my-system"
    }
  },
```

```
  "NonOdbcConnectors": {

    "admin@database-name": {
      "dbname": "my-netezza-database",
      "schema": "my-netezza-schema",
      "dialect": "netezza",
      "collibraSystemName": "my-system"
    },
    "admin@second-database-name": {
      "dbname": "my-second-netezza-database",
      "schema": "my-second-netezza-schema",
      "dialect": "netezza",
      "collibraSystemName": "my-system"
    }
  }
}
```

What's next

You can now prepare the rest lineage harvester configuration file and run it to create a technical lineage for Informatica PowerCenterSQL Server Integration ServicesIBM InfoSphere DataStage and, optionally, other data sources.

When you run the lineage harvester, the content in your local folder is sent to the Collibra Data Lineage server for processing.

> Note For more information about the scope, see the overview of supported data sources.

Download SQL files to the lineage harvester folder

You can download the SQL files of a data source that is stored locally and cannot be accessed via the network. The lineage harvester then stores the data source information in a ZIP file.

To create a technical lineage for these data sources, you only have to include the ID of the data source and the path to the ZIP file in the configuration file.

> Note Click here to see a list of all supported data sources.

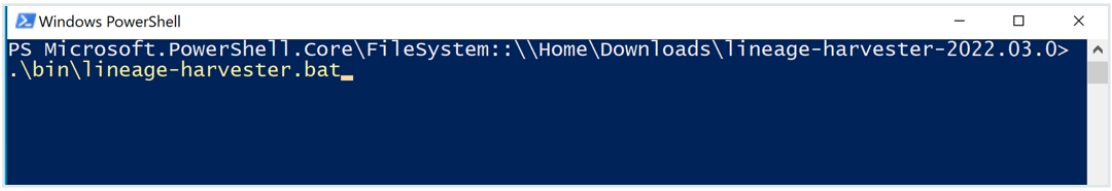
Prerequisites

- You have downloaded the lineage harvester and you have the necessary system requirements to run it.
- You have the necessary permissions to all database objects that the lineage harvester accesses.

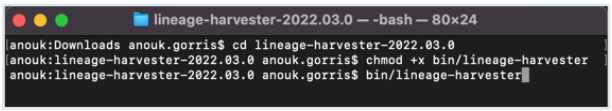
> Note For a detailed overview of the permissions that you need to access the data objects of your data sources, see the online user guide.

Steps

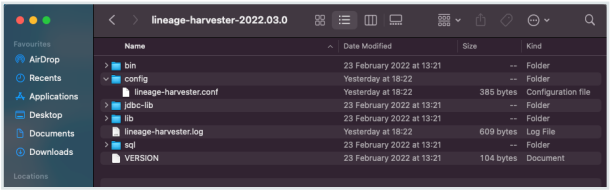
1. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



2. Save the configuration file in the **config** directory in the lineage harvester folder.
3. Prepare the configuration file.

> Tip Use the configuration file generator to easily create a configuration file.

4. When prompted, enter the passwords to connect to Collibra and your data sources. Do one of the following:

- Enter the passwords in the console.
  - » The passwords are encrypted and stored in **/config/pwd.conf**.
- Provide the passwords via command line.
  - » The passwords are stored locally and not in your lineage harvester folder.

5. Start the lineage harvester again and do one of the following:
   - To download the SQL files of all data sources in the configuration file, run the following command:

     ```
     ./bin/lineage-harvester load-sources
     ```

   - To download the SQL files of specific data sources in the configuration file, run the following command:

     ```
     ./bin/lineage-harvester load-sources -s "ID of the data source"
     ```

     > Tip This command allows you to download specific SQL files in the configuration file, without refreshing other SQL files. This reduces the time you need to download your SQL files, since you only download specific ones without affecting the others. If you want to download SQL files of multiple data sources, add `-s "ID of another data source"` per data source to the command.

     » The lineage harvester downloads the SQL files of the data sources and stores them in a ZIP file per data source in the lineage harvester output folder.

What's next?

You can now prepare a configuration file for theSQL files of data sources that you want to include in your technical lineage.

Prepare Informatica Intelligent Cloud Services <source ID> configuration file

You use the lineage harvester configuration file to access Informatica Intelligent Cloud Services Data Integration data objects. The lineage harvester processes the data objects to create a technical lineage. You also have to prepare a specific <source ID> configuration file that defines the Intelligent Cloud Services system name.

> Important You must prepare a <source ID> configuration file regardless of whether the `useCollibraSystemName` property in your lineage harvester configuration files is set to *true* or *false*.

> Tip The name <source ID> configuration file refers to the value of the `Id` property in the lineage harvester configuration file.

Prerequisites

You have Admin permission on all objects that you want to harvest.

Steps

1. Create a new JSON configuration file in the lineage harvester **config** folder.
2. Give the JSON file the same name as the value of the `Id` property in the lineage harvester configuration file.

   > Example The value of the `Id` property in the lineage harvester configuration file is `iics-source-1`. Therefore, the name of your JSON file should be *iics-source-1.conf*.

3. For each Informatica Intelligent Cloud Services connection, you can add the following content to the JSON file:

Property	Description
collibraSystemNames	This section contains the system information for Informatica Intelligent Cloud Services.
connections	This section contains the system connection information. This is required to reference to the system or server of the connection.
connectionName	The name of the connection.

Property	Description
collibraSystemName	The system or server name of the connection.
connectionDefinitions	This section contains the database, schema and dialect information for each connection in Informatica Intelligent Cloud Services. Note You can add connection information for each connection in the `connections` section.
connectionName	The name of the connection. The name must match with the name in a connection name in the `connections` section.
databaseName	The name of your database.
schemaName	The name of your schema.

Property	Description
dialect	The dialect of the connection. You can enter one of the following values: ◦ *bigquery* ◦ *db2* ◦ *hana* ◦ *hive* ◦ *greenplum* ◦ *mssql* ◦ *mysql* ◦ *netezza* ◦ *oracle* ◦ *postgres* ◦ *redshift* ◦ *snowflake* ◦ *spark* ◦ *teradata*

4. Save the configuration file.

Example of the <source-ID>.conf file

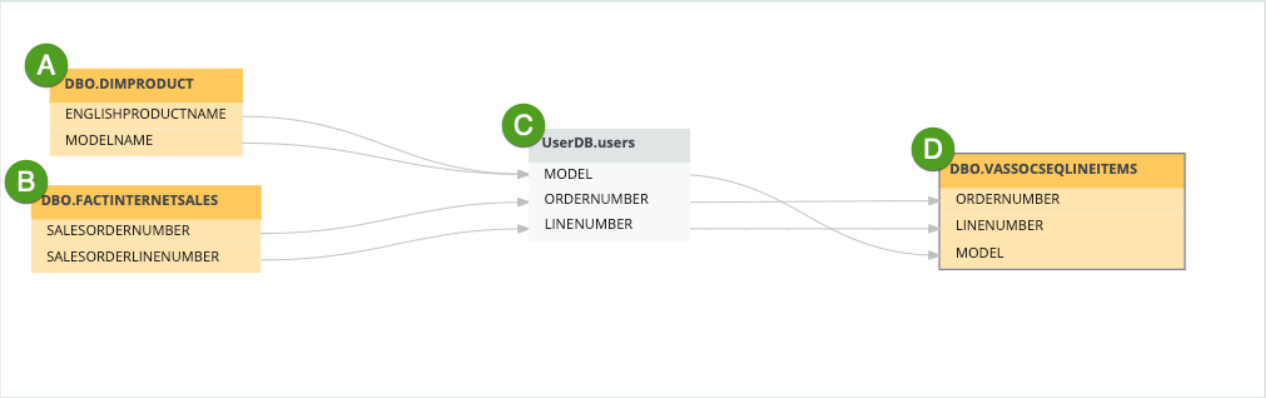
```
{
   "collibraSystemNames": {
       "connections": [
           {
               "connectionName": "DG_con_standby_cmdm_clientors",
               "collibraSystemName": "PUBLIC"
           },
           {
               "connectionName": "DG_con_dev_dg_dgiauser_su",
               "collibraSystemName": "PUBLIC"
           }
       ]
   },
   "connectionDefinitions": [
       {
           "connectionName": "DG_con_standby_cmdm_clientors",
           "databaseName": "main",
           "schemaName": "dbo",
```

```
            "dialect": "oracle"
        },
        {
            "connectionName": "DG_con_dev_dg_dgiauser_su",
            "databaseName": "main",
            "schemaName": "dbo",
            "dialect": "oracle"
        }
    ]
}
```

Custom technical lineage

You can create a custom technical lineage to include metadata of data sources that are not supported. To do so, you need to create and configure a JSON file that defines the custom technical lineage. You then add the properties of the JSON file to the configuration file.

Example
You want to create a technical lineage that shows relations between tables and columns from system A and system B to system C to system D (A and B -> C -> D). System A, B and D are supported data sources, but system C is a custom application. In this case, you can create a JSON file that contains the metadata of system C. This allows you to create a custom technical lineage that would be otherwise broken.



Create a custom technical lineage

You can create a custom technical lineage to include metadata of data sources that are not supported.

You can create two types of custom technical lineages:

- A simple custom technical lineage, which defines a basic object hierarchy and creates a lineage between two or more data objects.
- An advanced custom technical lineage, which contains a simple predefined technical lineage and defines transformations to create the lineage.

> Note In the local folder that you need to create, you can only have one JSON file. You can, however, add other files in the harvested directory and subdirectories and refer to those files from within the JSON file.

Prerequisites

- You have downloaded the lineage harvester and you have the necessary system requirements to run it.
- You have the necessary permissions for all database objects that the lineage harvester accesses.
- You have prepared the physical data layer in Data Catalog.

  > Note To stitch the data objects of data sources mentioned in the JSON file with Data Catalog assets, you first have to register those data sources in Data Catalog and you have to use a structure that matches the structure of ingested assets in Data Catalog.

Create a simple custom technical lineages

1. Create a local folder.
2. Create a new JSON file in the local folder.

3. Name the JSON file *lineage.json*.
4. Add the following mandatory sections to the JSON file:

Properties	Description
version	The version of the JSON architecture. Note Currently, you can only use version 1.0.
tree	This section contains tree definitions of data objects between which lineages can be defined. Each node of a tree contains the name, type and optionally children or leaves properties which form a hierarchy of data objects. You can reuse the same properties in one node to map all data objects in the hierarchy. Tip Usually, the structure you map is the following: system > database > schema > table > column. The system is optional, unless the `useCollibraSystemName` property is set to `true` in the lineage harvester configuration file. The Collibra Data Lineage can stitch these data objects to assets in Data Catalog. However, you can also map custom objects, for example dashboards and reports. Custom objects cannot be stitched to assets in Data Catalog.
name	The name of your data object. This is the system name, database name, schema name, table name or column name. Warning ◦ The names are case sensitive. ◦ The names of data objects of the same type must be unique.

Properties	Description
type	The type of your data object. For example: `system`, `database`, `schema`, `table` or `column`.
children	The sub-objects that have a hierarchical relation to the defined data object. Each child also has the `name` and `type` properties and can have children of its own, except for the penultimate child which has leaves instead of children. Leaves are children without children. > Note Use the children property to define sub-objects, but use the leaves property if the object is on the penultimate level. For example, to define columns that have a relation to a table node.
leaves	The sub-objects of another sub-object that is defined in a `children` property, but cannot have sub-objects of their own. > Note Technical lineage only shows relations between leaf nodes of the tree. Leaves are usually columns that have a relation to a table node in the tree structure.

Properties	Description
lineages	This section contains the path from a source to a target and defines the mappings and transformations that should be processed by the Collibra Data Lineage server. Note If you create a lineage between data objects that are also assets in Data Catalog, the Collibra Data Lineage server automatically stitches the data objects to the assets in Data Catalog. However, you can also create a lineage between custom data objects that are not assets in Data Catalog, for example reports and dashboards.
src_path	The hierarchical path to the source data object. This data object is shown as a leaf in the tree node.
<data objects>	All data object names in the hierarchical path to the source leaf. Example of data objects that can be stitched: system > database > schema > table > column. Example of data objects that cannot be stitched: dashboard > report > column.
trg_path	The hierarchical path to the target data object. This data object is shown as a leaf in the tree node.
<data objects>	All data object names in the hierarchical path to the target leaf. Example of data objects that can be stitched: system > database > schema > table > column. Example of data objects that cannot be stitched: dashboard > report > column.

Properties	Description
mapping	The mapping name. This refers to the queries used in the technical lineage.
source_code	The transformation code. This determines how the technical lineage is constructed. Tip The source code can be a SQL statement or code that manipulates data.

5. In your configuration file, add the path to the JSON file.

Example of a JSON file for a simple custom technical lineage

```
{
  "version": "1.0",
  "tree": [
    {
      "name": "UserDB",
      "type": "database",
      "children": [
        {
          "name": "SCH",
          "type": "schema",
          "children": [
            {
              "name": "users",
              "type": "table",
              "leaves": [
                {
                  "name": "membership_type",
                  "type": "column"
                }
              ]
            }
          ]
        }
      ]
    },
    {
      "name": "User dash",
      "type": "dashboard",
      "children": [
        {
```

```
          "name": "Memberships",
          "type": "report",
          "leaves": [
            {
              "name": "Type",
              "type": "column"
            }
          ]
        }
      ]
    }
  ],
  "lineages": [
    {
      "src_path": [
        {"database": "UserDB"},
        {"schema": "SCH"},
        {"table": "users"},
        {"column": "membership_type"}
      ],
      "trg_path": [
        {"dashboard": "User dash"},
        {"report": "Memberships"},
        {"column": "Type"}
      ],
      "mapping": "make_report",
      "source_code": "report = rep(data)"
    }
  ]
}
```

Create an advanced custom technical lineage

1. Create a local folder.
2. Create a new JSON file in the local folder.
3. Name the JSON file *lineage.json*.
4. In the same local folder, store all of the source codes that you want to reference in the JSON file.
5. Add the following sections to the JSON file:

Properties	Description
version	The version of the JSON architecture. Note Currently, you can only use version 1.0.

Properties	Description
tree	This section contains tree definitions of data objects between which lineages can be defined. Each node of a tree contains the name, type and optionally children or leaves properties which form a hierarchy of data objects. You can reuse the same properties in one node to map all data objects in the hierarchy. Tip Usually, the structure you map is the following: system > database > schema > table > column. The system is optional, unless the `useCollibraSystemName` property is set to `true` in the lineage harvester configuration file. The Collibra Data Lineage can stitch these data objects to assets in Data Catalog. However, you can also map custom objects, for example dashboards and reports. Custom objects cannot be stitched to assets in Data Catalog.
name	The name of your data object. This is the system name, database name, schema name, table name or column name. Warning ◦ The names are case sensitive. ◦ The names of data objects of the same type must be unique.
type	The type of your data object. For example: `system`, `database`, `schema`, `table` or `column`.

Properties	Description
children	The sub-objects that have a hierarchical relation to the defined data object. Each child also has the `name` and `type` properties and can have children of its own, except for the penultimate child which has leaves instead of children. Leaves are children without children. Note Use the children property to define sub-objects, but use the leaves property if the object is on the penultimate level. For example, to define columns that have a relation to a table node.
leaves	The sub-objects of another sub-object that is defined in a `children` property, but cannot have sub-objects of their own. Note Technical lineage only shows relations between leaf nodes of the tree. Leaves are usually columns that have a relation to a table node in the tree structure.
lineages	This section contains the path from a source to a target and defines the mappings and transformations that should be processed by the Collibra Data Lineage server. Note If you create a lineage between data objects that are also assets in Data Catalog, the Collibra Data Lineage server automatically stitches the data objects to the assets in Data Catalog. However, you can also create a lineage between custom data objects that are not assets in Data Catalog, for example reports and dashboards.

Properties	Description
src_path	The hierarchical path to the source data object. This data object is shown as a leaf in the tree node.
<data objects>	All data object names in the hierarchical path to the source leaf. Example of data objects that can be stitched: system > database > schema > table > column. Example of data objects that cannot be stitched: dashboard > report > column.
trg_path	The hierarchical path to the target data object. This data object is shown as a leaf in the tree node.
<data objects>	All data object names in the hierarchical path to the target leaf. Example of data objects that can be stitched: system > database > schema > table > column. Example of data objects that cannot be stitched: dashboard > report > column.
mapping_ref	The mapping of the source codes that are located in the same directory of the JSON file and their positions in the technical lineage. Note The positions are zero based. The first character in a sequence has position 0.

Properties	Description
source_code	The source code for which you provide a path in the `codebase_files` node. Tip The source code can be a SQL statement or code that manipulates data.
mapping	The mapping name of the mapping defined in the `codebase_files` node.
codebase_pos	The positions of a source code file that is located in the same directory of the JSON file. These source code positions will be highlighted under the technical lineage of a column.
codebase_files	This section defines the reference to source code files that are stored in the same directory as the JSON file.
<source code path>	The reference to source code that is located in the same directory as the JSON file. This contains mappings of the source codes and their positions.
mapping_refs	The mapping of the source code for which you provided the path in <source code path>.

6. In your configuration file, add the path to the JSON file.

Example of a JSON file for an advanced custom technical lineage

```
{
  "version": "1.0",
  "tree": [
    {
      "name": "UserDB",
      "type": "database",
      "children": [
        {
```

```
          "name": "SCH",
          "type": "schema",
          "children": [
            {
              "name": "users",
              "type": "table",
              "leaves": [
                {
                  "name": "membership_type",
                  "type": "column"
                }
              ]
            }
          ]
        }
      ]
    },
    {
      "name": "User dash",
      "type": "dashboard",
      "children": [
        {
          "name": "Memberships",
          "type": "report",
          "leaves": [
            {
              "name": "Type",
              "type": "column"
            }
          ]
        }
      ]
    }
  ],
  "lineages": [
    {
      "src_path": [
        {"database": "UserDB"},
        {"schema": "SCH"},
        {"table": "users"},
        {"column": "membership_type"}
      ],
      "trg_path": [
        {"dashboard": "User dash"},
        {"report": "Memberships"},
        {"column": "Type"}
      ],
      "mapping_ref": {
        "source_code": "user_utils.py",
        "mapping": "showUserData",
        "codebase_pos": [
          {
            "pos_start": 25,
```

```
            "pos_len": 31
          }
        ]
      }
    }
  ],
  "codebase_files": {
    "user_utils.py": {
      "mapping_refs": {
        "showUserData": {
          "pos_start": 0,
          "pos_len": 56
        }
      }
    }
  }
}
```

What's next

When you're done configuring the JSON file, you can prepare the lineage harvester configuration file and enter the correct properties to create a technical lineage using the JSON file.

Manage technical lineage ingestion

You can manage which data objects, for example columns and tables, are ingested in the technical lineage. By doing this, you can decide:

- Which data objects you want to visualize in the technical lineage.
- Between which columns you want to create new relations of the type "Data Element targets / sources Data Element" into Catalog.

You manage technical lineage ingestion by excluding data objects from the technical lineage ingestion or change queries that are used to extract data from the database.

> Note
> - If you change queries, you can only use supported SQL syntax
> - The lineage harvester does not support proxy server authentication, but you can manually connect to a proxy server via command line. For more information, see Connecting to a proxy server.

Steps

1. Open the lineage harvester folder.
2. Go to the **sql** folder and open the folder of the data source type of which you want to exclude tables or schemas or change queries.
3. Create a copy of the file you want to edit.
4. Rename the copy to *[original name]-custom.sql*.

   > Example You want to change the file columns.sql, so you name the copy of this file and rename it to *columns-custom.sql*.

5. Delete or edit the content of the new SQL file to include or exclude specific tables or schemas or change specific queries in the file.
6. Save the new SQL file.
   » The lineage harvester uses the new file and ignores the old one.

Schedule jobs

You can use Task Scheduler on Windows or Crontab on Mac and Linux to make the lineage harvester run scheduled jobs at specific times, dates or intervals. In a scheduled job, the lineage harvester uploads data source information to the Collibra Data Intelligence Cloud and Data Catalog automatically creates new relations of the type "Data Element sources / targets Data Element"

- Between data objects in your data source and assets from registered data sources.
- Between ingested assets from BI sources and Data Catalog assets from registered data sources.

You can run one scheduled job for each data source that is listed in the same configuration file.

> Note If you provide the passwords to your Collibra environment and/or to your individual data sources via stdin, you have to use the correct command.

> Example You created a configuration file with two data sources. Data source A can run a scheduled job each day at 11 pm, while data source B can run a scheduled job every two days at 6 am.

Delete the technical lineage of a data source

You can delete the technical lineage of a data source if you no longer want to see it in the technical lineage graph.

> Note You always need at least one source in your lineage harvester configuration file.

Prerequisites

- You have a global role that has the Manage all resources global permission.
- You have a global role with the Technical lineage global permission.
- You have downloaded the lineage harvester and you have the necessary system requirements to run it.
- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.
- You have added Firewall rules so that the lineage harvester can connect to:
  - All Collibra Data Lineage servers within your geographical location:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)
  - The host names of all databases in the lineage harvester configuration file.

Steps

1. In the lineage harvester folder, open your lineage harvester configuration file.
2. Delete the section with connection properties of the data source you no longer want to see a technical lineage for.
3. Save the configuration file.
4. Start the lineage harvester again in the console and run the following command:
   - for Windows: `.\bin\lineage-harvester.bat full-sync`
   - for other operating systems: `./bin/lineage-harvester full-sync`
5. When prompted, enter the password to connect to your Collibra Data Intelligence Cloud and data sources in the configuration file.
   » The lineage harvester uploads the metadata of the remaining data sources in the configuration file to the Collibra Data Lineage server.
   » The Collibra Data Lineage server synchronizes the technical lineage and removes the deleted data source from the technical lineage graph.

Technical lineage viewer

The technical lineage viewer shows the technical lineage and allows you to edit the view. You can access the technical lineage viewer via the Technical lineage tab on Column and Table asset pages and BI assets of the same level.

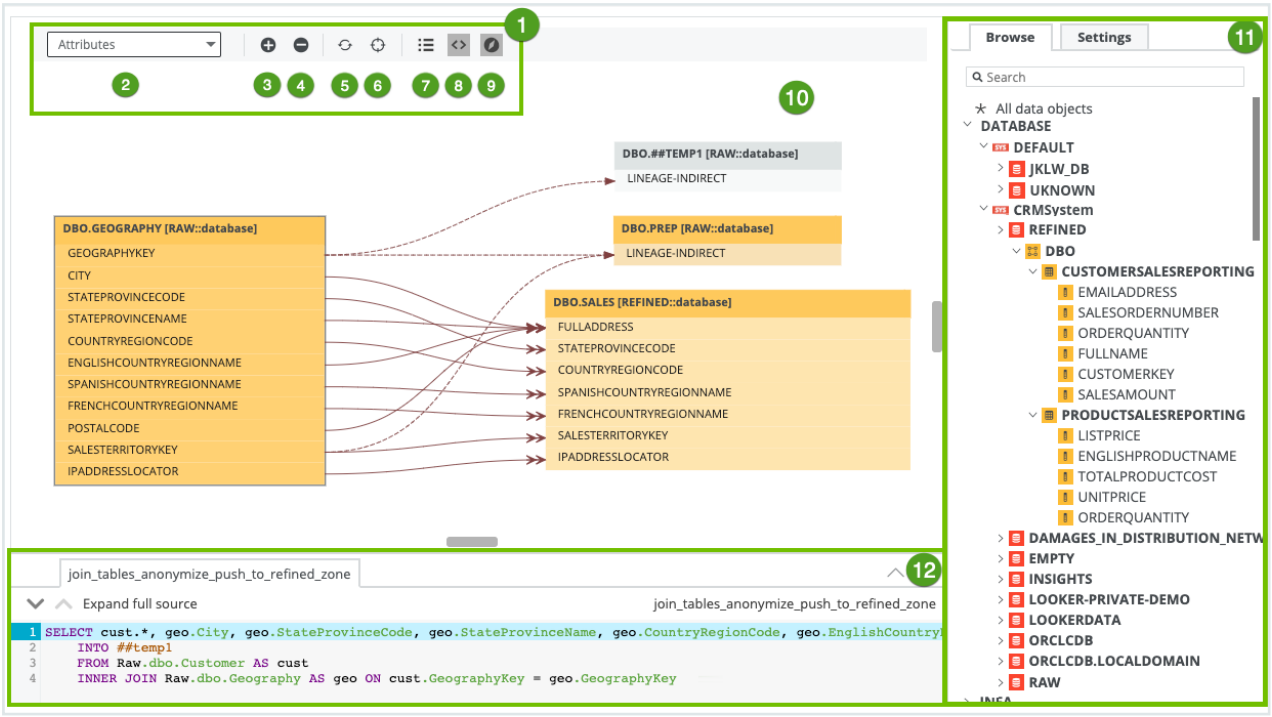
> Tip For more information about the technical lineage for Looker or Power BI, we highly advise you to read the dedicated sections in the user guide.

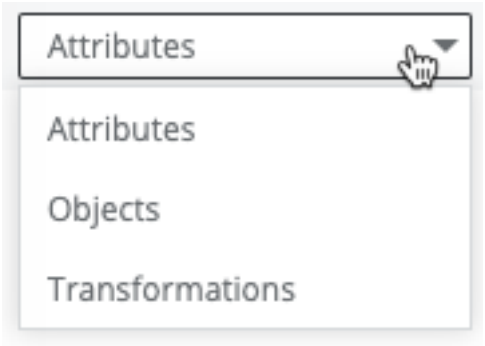
Technical lineage tab

You can only see the Technical lineage tab on a Column or Table asset page when you have the following prerequisites:

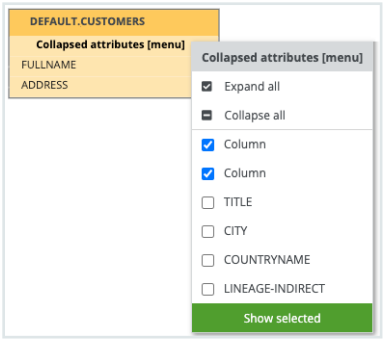
- You have a global role with the Catalog global permission, for example Catalog Author.
- You have a global role with the Technical lineage global permission.

Technical lineage viewer

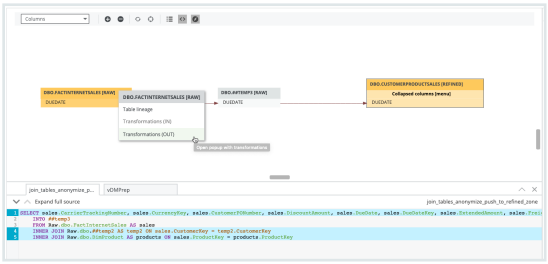


No	Name	Description
1	Toolbar	The toolbar to work with technical lineage. The toolbar helps you to edit basic settings that apply to the entire lineage.
2		Drop-down list to determine which details (attributes, objects or trans-formations) you want to show in the technical lineage graph.
3	⊕	Button to zoom in on the technical lineage.

No	Name	Description
4		Button to zoom out on the technical lineage.
5		Button to refresh the technical lineage. This discards all the changes that you made to the technical lineage and restores it to the initial state.
6		Button to reposition the technical lineage to the starting position.
7		Button to show or hide the legend panel.
8		Button to show or hide the source code pane.
9		Button to show or hide the **Browse** and **Settings** tab panes.

No	Name	Description
10	Technical lineage graph	The actual visualization of the traceability of the current data object, according to your selection in the **Browse** tab pane. If you select a specific column in a table with multiple columns, you can click **Collapsed columns [menu]** to show all other columns, collapse all columns or only show selected columns in the same table.  Tip Data objects that are stitched to assets in Data Catalog have a yellow background. Other data objects that the lineage harvester collected from your data source, but are not stitched and therefore are not assets in Data Catalog, have a gray background.

No	Name	Description
11	Tab panes	Tab panes that contain useful tools to browse through your technical lineage or determine which content is visualized in the technical lineage.
	Browse tab pane	This pane can be used to search for specific data objects or show statistics on the amount of tables and views in use. More information.
	Settings tab pane	This pane can be used to search for transformation code, edit the visualization of the technical lineage, see the status of the source code, check the stitching results or export your technical lineage to PDF, PNG or CSV. More information.

No	Name	Description
12	Source code pane	The source code pane shows the source code of specific data objects. It can be used to easily find issues in the data flow. The source code pane is shown when you click <> in the toolbar or when you right-click a column or table and click **Transformations (IN)** or **Transformations (OUT)** which shows the transformation logic in the source code pane. 

The technical lineage graph

The technical lineage graph consists of nodes and edges. Each node represents a corresponding object in a data source. Each edge shows a relation between nodes.

Nodes and edges in the technical lineage graph show how data flows from source to destination. Understanding the nodes and edges better, enriches your technical lineage experience.

Consider the following visual elements in the technical lineage graph:

- Relation types
- Messages
- Colors
- Icons
- Arrows

- Collapsed attributes menu
- Right-click menu

Relation types

The technical lineage graph shows relations between columns in the graph. The Collibra Data Lineage creates and shows the following relation type between stitched assets and other data objects:

Head	Role	Co-role	Tail	ID
Data Element	targets	sources	Data Element	00000000-0000-0000-0000-000000007069

Messages

The technical lineage graph might show different messages to alert you. The following messages are the most common:

Message	Description
Edges count exceeds the limit 1000.	The technical lineage graph exceeds the limit of 1000 nodes and is too large to display. This happens, for example, if you have a table with many columns and you try to show the technical lineage of all columns in a table in one graph. Note You cannot manually change this limit.
The current asset doesn't have a technical lineage yet.	This message is shown if you didn't create a technical lineage for the data source of the asset. Use the Browse tab pane to navigate through the data object for which a technical lineage graph is available.

Message	Description
Technical lineage cannot be shown.	The technical lineage graph cannot be shown, because there are too many data objects. This happens, for example, when you created a technical lineage for multiple data source and you click **All data objects** in the Browse tab pane. Use the Browse tab pane to view specific parts of the technical lineage graph or click the suggested data objects to see their graph.

Colors

The technical lineage graph shows different colors to indicate which data objects are stitched to assets in Data Catalog and which are not.

Background colors

The background color of a node indicates whether or not the data object was stitched to an asset in Data Catalog, and whether something went wrong.

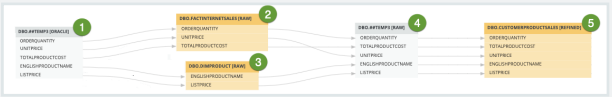
A node has one of three background colors:

Color	Description
Yellow	Data objects from your data source that are stitched to assets in Data Catalog

Color	Description
Gray	Data objects, for example temporary tables and columns, that the lineage harvester collects from your data sources, but are not stitched to assets in Data Catalog. Warning We do not support stitching for Looker assets. We do support stitching for Power BI assets, but the stitched assets still have a gray background. This is a known issue.
Red	Attributes that are automatically assigned to a data object, because of missing DDL statements. If you want to remove objects with a red background, change the statements and rerun the lineage harvester.

Since a technical lineage shows how data flows from source to destination, it is possible to see a lineage graph with both yellow, red and gray nodes.

Example The following technical lineage graph shows two nodes with a gray background and three nodes with a yellow background. Node 1 and 4 contain data objects that are not stitched to assets in Data Catalog while nodes 2, 3 and 5 contain existing assets in Data Catalog that were stitched to the corresponding data objects when you created the technical lineage.



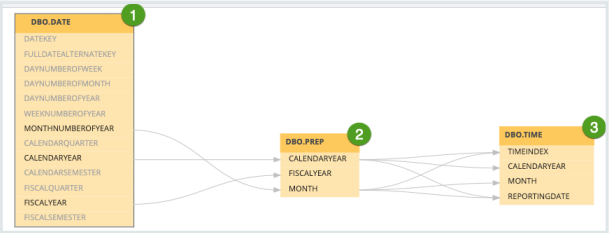
Font colors

The font color of data objects in the technical lineage graph indicates whether or not there is a relation between this data object and one or more other data objects.

A node has one of two font colors:

Color	Description
Black	At least one direct or indirect relation exists between the data object and another.
Gray	No relation exists between the data object and another.

Example The following technical lineage graph shows three nodes. The node 1 contains data objects that have no incoming or outgoing edges to other data objects in the technical lineage. Nodes 2 and 3 only contain data objects that have a relation to other data objects in the technical lineage.



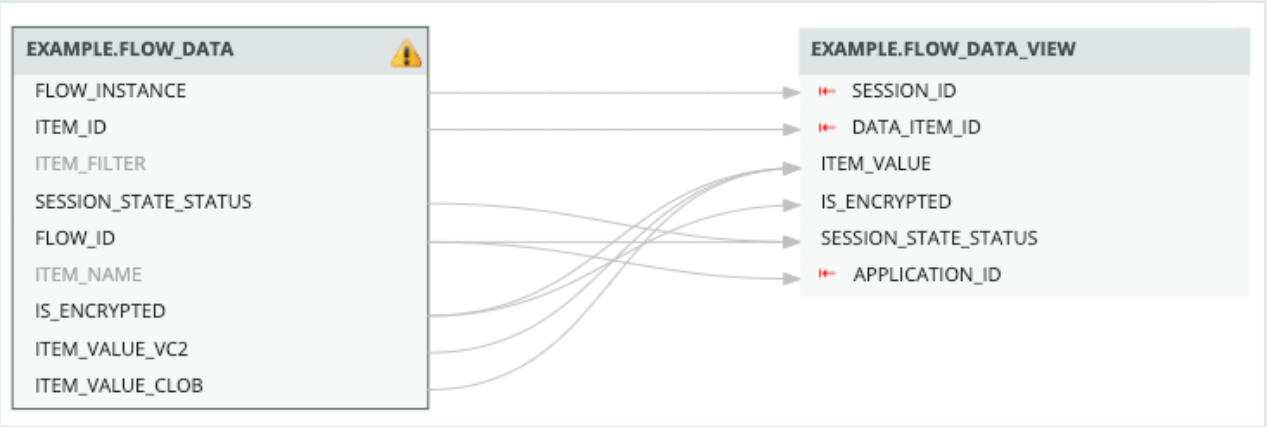
Icons

Collibra uses various icons in the technical lineage graph.

Icon	Description
⚠	The name of a table was found by the full-text search in the source code on which the analysis failed. Consequently, the lineage flow of the table is probably incomplete. If you click **Show failed SQLs** on the right click menu of the table, the failed SQL queries appear in the source code pane at the bottom of the page.
↺	The lineage is cyclic, for example A → B → C → A. It only appears if you enabled the only ending points option in the **Settings** tab pane.

Icon	Description
⇤	A relation for the data objects exists, but it isn't shown, for example because you set the technical lineage flow depth to a lower value than the actual graph size.

Example The following Technical lineage graph shows two nodes. The first node has an icon to indicate that the lineage flow you currently see is probably incomplete. The second node has three data objects that have a relation to other data objects, but the edges that represent that relation are not shown.



Arrows

Arrows are incoming or outgoing edges that show how the data flows from source to destination. They represents relations of the type "Data Element sources / targets Data Element".

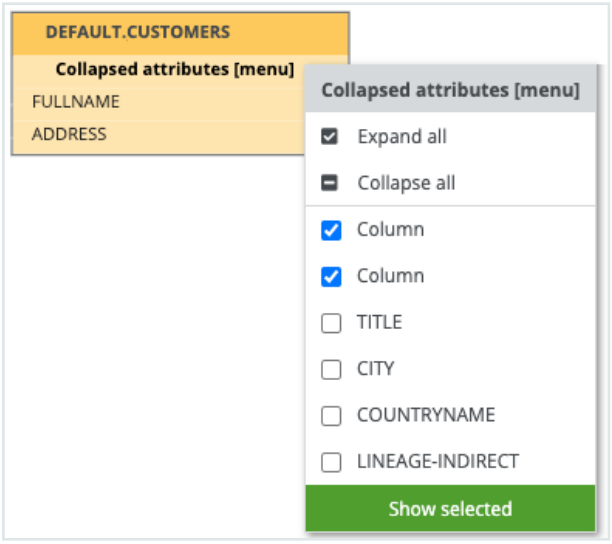
There are two ways in which an arrow can be shown:

Arrow type	Description
Single	Shows the full lineage without skipping certain data objects.
Double	Shows that there are hidden data objects in the technical lineage graph. This happens when only the endpoints of the technical lineage flow are shown.

Example The following Technical lineage graph shows three nodes. Edges with double arrows are shown between node 1 and 3. These edges indicate that there are other nodes between these nodes in the full technical lineage flow. Node 2 has outgoing edges with single arrows. These edges indicate that there is a direct relation between node 2 and 3.

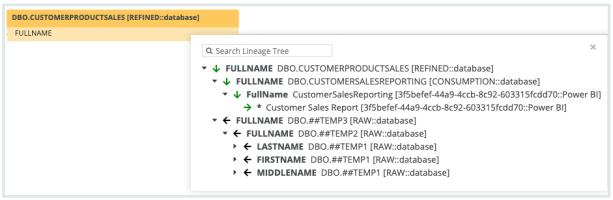
Collapsed attributes menu

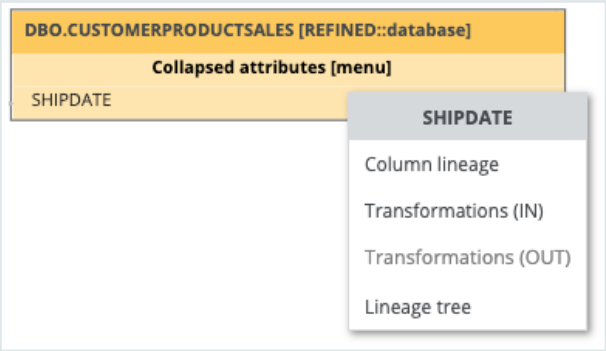
If you select a specific column in a table with multiple columns, you can click **Collapsed attributes [menu]** to show all columns, collapse all columns or only show selected columns in the same table.



Right-click menu

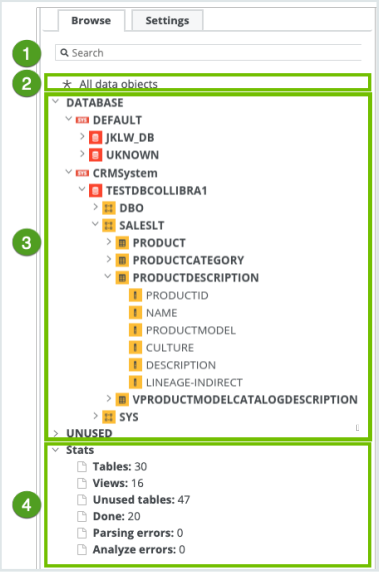
If you right-click a node, you can perform several specific actions on that node.

Functionality	Description
Column/Table lineage	Switch to the technical lineage graph of the selected column or table.
Transformation (IN)	Show the transformation logic of the incoming source code fragments in the source code pane.
Transformation (OUT)	Show the transformation logic of the outgoing source code fragments in the source code pane.
Lineage tree	Show an alternative way to view the flow of data objects, called the lineage tree. The lineage tree is particularly useful if there are many nodes in a lineage. It enables you to see the entire lineage in one pop-up, which means you no longer have to scroll through the technical lineage graph to see the full lineage. The lineage tree uses arrows to visualize the traceability of data objects: • Green arrows represent outgoing edges. • Black arrows represent incoming edges. 
Custom features	When the lineage flow of the table is incomplete or there is an issue in the source code of a data object, the right-click menu shows the **Show failed SQLs** option. If you click this option, the source code pane opens and shows the SQL queries that failed.



Technical lineage Browse tab pane

The **Browse** tab pane allows you to navigate to and search for a specific data object within the technical lineage tree.



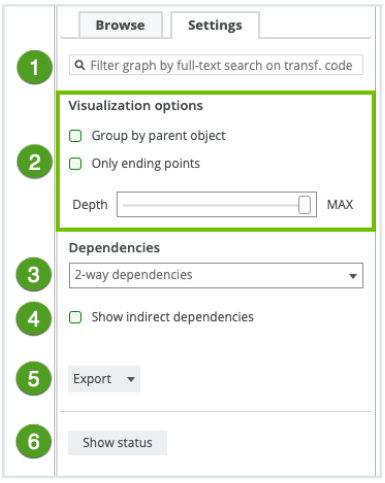
No	Name	Description
1	Search	A search field that you can use to find a specific data object.
2	All data objects	A link to the complete technical lineage, showing all data objects in your data sources.

No	Name	Description
3	Navigation tree	A navigation tree in which you can search for specific data objects and visualize them in your technical lineage. The data objects are grouped by node type and have the following structure: system (if applicable) > database > schema > table > column. Note The list of data objects contains all systems, databases, schemas, tables and columns that were collected from the data sources by the lineage harvester. If available, it also shows the technical lineage of BI sources, for example Power BI and Looker. In that case, the structure follows the existing structure in the BI source metadata.

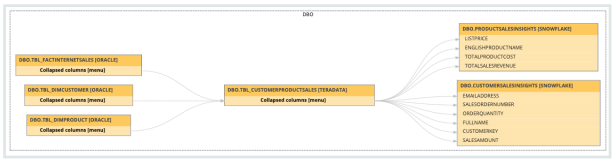
No	Name	Description
4	Stats	Statistics that show which information is or is not visualized in the technical lineage. The statistics contain the following data: • **Tables**: the amount of tables that are shown in the technical lineage. • **Views**: the amount of views that are shown in the technical lineage. • **Unused tables**: the amount of tables in your data source that are not shown in the technical lineage. > Tip This metric is hidden when there are no unused tables. • **Unused views**: the amount of views in your data source that are not shown in the technical lineage. > Tip This metric is hidden when there are no unused views. • **Done**: the amount of queries that were processed successfully. • **Parsing errors**: the amount of queries with invalid or unidentified syntax. • **Analyze errors**: the amount of columns that are not linked to a table.

Technical lineage Settings tab pane

The **Settings** tab pane allows you to edit the technical lineage, search for queries and export the technical lineage.



No	Name	Description
1	Search field	A search field to find specific transformation code in a selected object or attribute. As you type, corresponding object names from the technical lineage appear in a drop-down list. If you press `Enter`, the technical lineage only shows the parts that contain your search word(s).

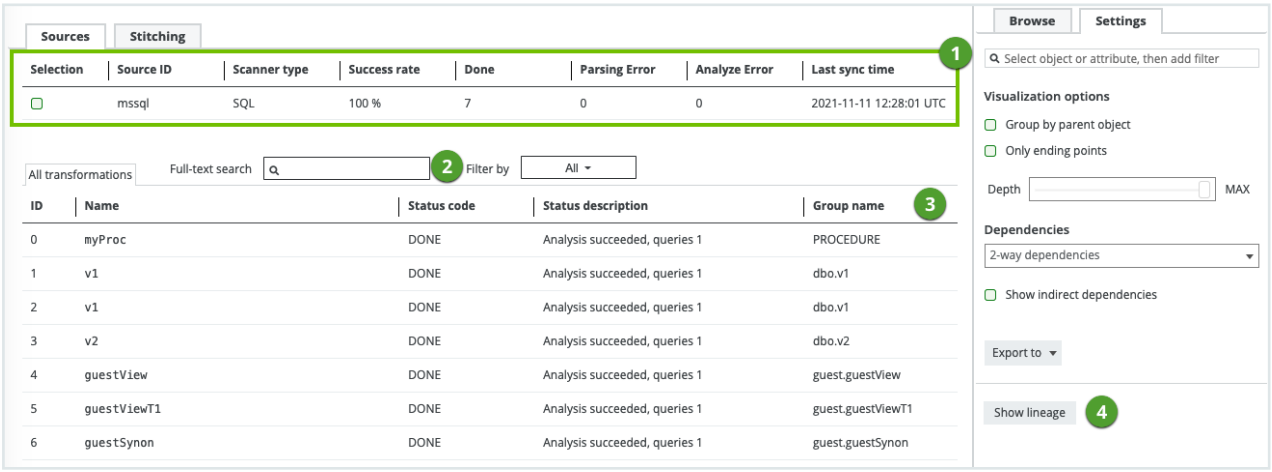
No	Name	Description
2	Visualization options	Options to define how you will see the data objects in the technical lineage.
	Group by parent object	Option to group tables and columns together by their hierarchical parent object. Example A schema is the parent object of a table. 
	Only ending points	Option to hide all data objects in the middle of the data flow and only show the ending points of the technical lineage.
	Depth	A slider that determines the maximum flow depth. The relation path length from the first node in the technical lineage graph to any other node is automatically adjusted to the maximum flow depth. If you see in the technical lineage graph, the flow depth is set to a lower value than the actual graph size.
3	Dependencies	Drop-down to select the dependencies that you want to visualize. You can select one of the following dependencies: • Inbound dependencies only • Outbound dependencies only • 2-way dependencies
4	Show indirect dependencies	Option to enable indirect dependencies.

No	Name	Description
5	Export	Button to export your technical lineage to PDF, PNG or CSV.
6	Show status	Button to switch to the Sources tab page, which shows the analysis log files of your data sources and the Stitching tab page, which shows an overview of assets and data objects and shows which are stitched.

Technical lineage Sources tab page

When you create a configuration file and run the `full-sync` command in the lineage harvester, your data sources are uploaded to the Collibra Data Lineage server to be analyzed and processed. The Sources tab page shows the transformation details or source code that was analyzed and the results of this analysis.

You can access the Sources tab page by clicking **Show status** on the Settings tab pane.



No	Name	Description
1	Summary per data source	A summary per data source. You can also select data sources to filter the results.
	Selected	Checkboxes to filter on a data source in the transformations table. If you select none, the transformations table contains all transformations.
	Source ID	The ID of your data source. You entered this ID in the configuration file.
	Scanner type	The type of scanner that is used to scan the queries in your data source.
	Success rate	The success rate of the data source analysis on the Collibra Data Lineage server. The success rate indicates how complete your technical lineage is. Important The success rate of a technical lineage gives a good indication of the processing success. A success rate less than 100%, however, does not mean processing was unsuccessful. A parsing error, for example, which negatively affects the success rate, does not always negatively affect the completeness of the lineage.
	Done	The amount of queries that were scanned and analyzed.
	Parsing Error	The amount of parsing errors.
	Analyze Error	The amount of analysis errors.
	Last sync time	The last time the data source was uploaded to the Collibra Data Lineage server, for analysis and processing.

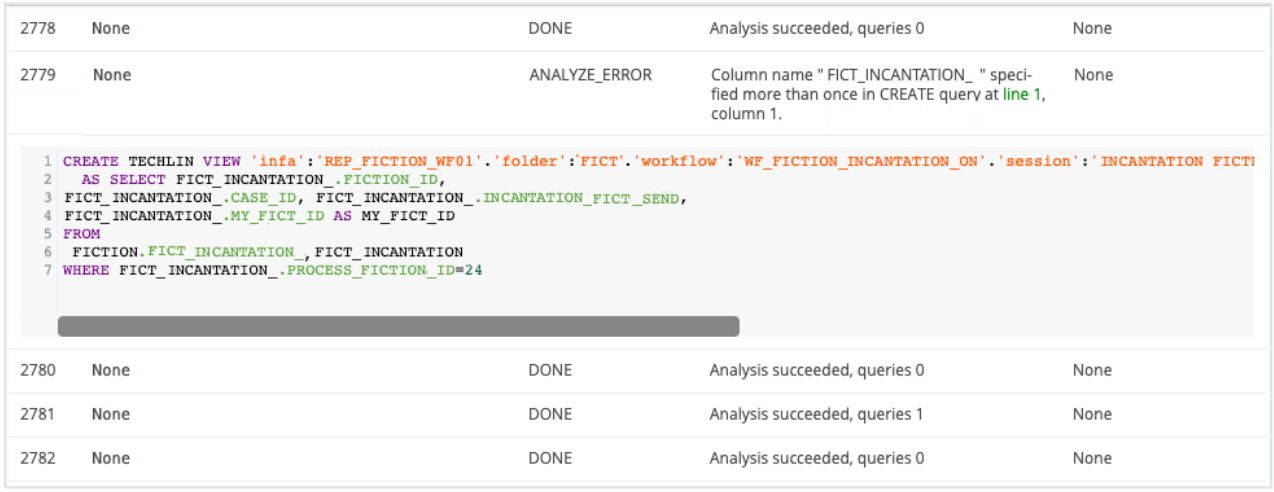
No	Name	Description
2	Search tools	Tools to help you search for specific source code fragments.
	Full-text search	A search field to find specific queries in the log files. Type what you are looking for and press `Enter`.
	Filter by	A drop-down list to filter the source codes based on their status code.

No	Name	Description
3	Transformations table	The table that contains details of the transformations and source code (fragments). You can filter the rows in the table by selecting data sources in the data source table and by using the search tools. Tip If you click a source code fragment, you can see the log file attached to it.
	ID	The ID of the source code fragments or transformation details, which are assigned in chronological order.
	Name	The name of the specific source code fragment or transformation detail. You can also see the source code fragment name in the source code pane in the technical lineage graph.
	Status code	The status of the analysis. A source code fragment or transformation detail can have one of the following status codes: • `DONE`: All queries are processed successfully. • `ERROR`: Some queries could not be processed. • `PARSING_ERROR`: The syntax of some queries is invalid or unidentified. • `ANALYZE_ERROR`: Some columns are not linked to a table.
	Status description	The description of the status code that provides more information about the analysis and shows how many queries were processed.

No	Name	Description
	Group name	The name of the package or procedure to which the source code fragment or transformation details belongs.
4	Show lineage	Button to go back to the technical lineage graph.

Analysis results

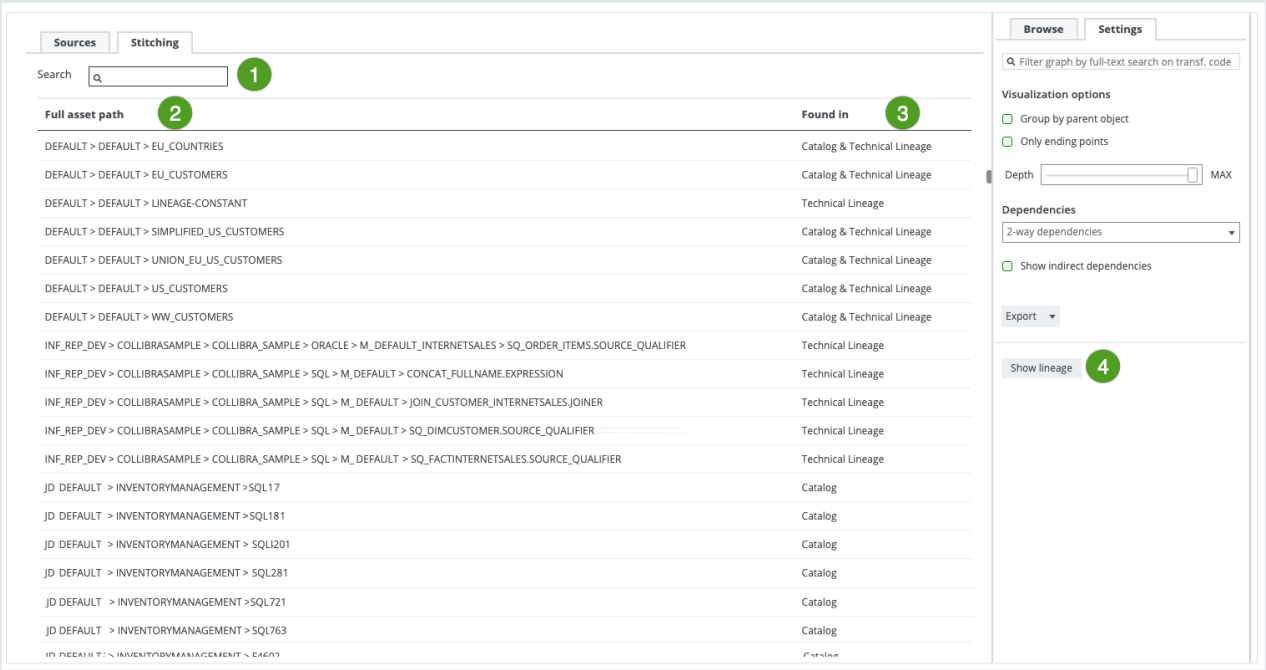
If you click one of the rows in the Transformations table, a file with the analysis results attached to the source code or transformation details opens. You can use these files to easily find errors in the source code or transformation details of your data source.



Technical lineage Stitching tab page

The Stitching page shows the full path of assets in Data Catalog and data objects of the data sources for which you created a technical lineage. You can use it to easily check which assets are stitched and which are not.

You can access the Stitching tab page by clicking **Show status** on the Settings tab pane.



No	Name	Description
1	Search field	A search field to find specific assets or data objects. Type what you are looking for and press `Enter`.
2	Full asset path	The full path to all data objects on the Collibra Data Lineage server and all assets in Data Catalog.

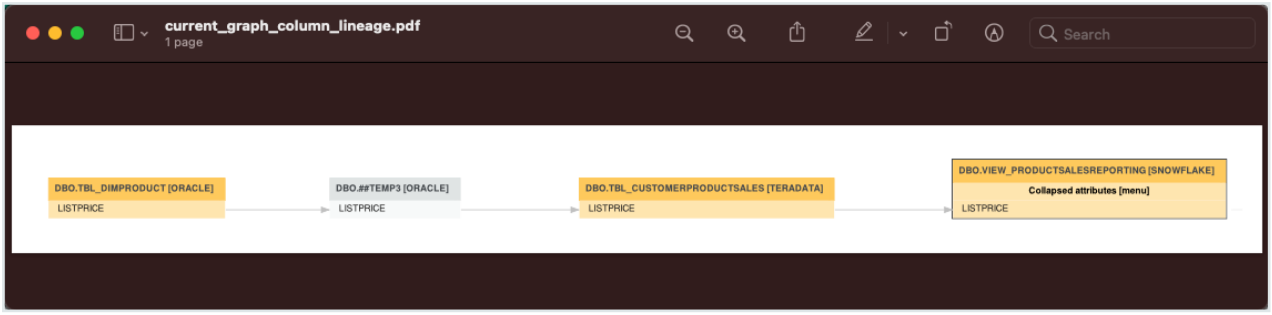
No	Name	Description
3	Found in	The location where the asset or data object was found. There are three possible locations: • **Data Catalog**: The asset was found in Data Catalog, but it does not match the full path of a data object on the Collibra Data Lineage server. As a result, there is no technical lineage created for this asset. • **Technical lineage**: The data object was found in the data source for which you created a technical lineage, but it does not match the full path of an asset in Data Catalog. As a result, the data object is shown in technical lineage with a gray background. • **Data Catalog & Technical lineage**: An asset and a data object with the same full path were found in Data Catalog and on the Collibra Data Lineage server. As a result, they were stitched and are shown in technical lineage with a yellow background.
4	Show lineage	The button to go back to the technical lineage graph.

Technical lineage export types

If you want to share a technical lineage graph of your technical lineage, you can export the information to PDF, PNG or CSV, via the Settings tab pane.

PDF and PNG exports

The PDF and PNG exports show only the technical lineage graph of the selected table or column.



CSV export

The CSV export option generates a ZIP file with the following CSV file:

File name	File content
current_graph_column_lineage.csv	The technical lineage graph of the selected column or table.

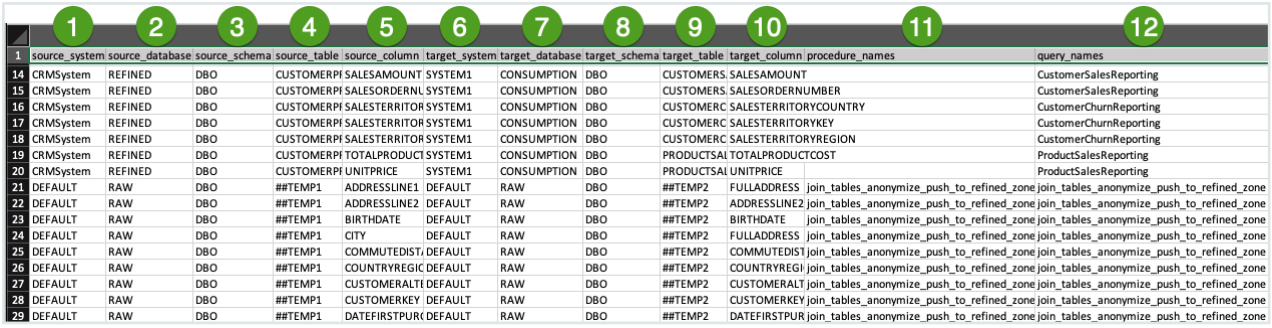
Full CSV

The Full CSV option generates a ZIP file with the following CSV files:

File name	File content
current_graph_column_lineage.csv	The technical lineage graph of the selected column or table.
full_batch_column_lineage.csv	The technical lineage graph of the full technical lineage.

Example

The current_graph_column_lineage CSV file and the full_batch_column_lineage CSV files show the same information, but with a different scope. These files show how data flows from source to target.



	1	2	3	4	5	6	7	8	9	10	11	12
1	source_system	source_database	source_schema	source_table	source_column	target_system	target_database	target_schema	target_table	target_column	procedure_names	query_names
14	CRMSystem	REFINED	DBO	CUSTOMERPI	SALESAMOUNT	SYSTEM1	CONSUMPTION	DBO	CUSTOMERS	SALESAMOUNT		CustomerSalesReporting
15	CRMSystem	REFINED	DBO	CUSTOMERPI	SALESORDERNU	SYSTEM1	CONSUMPTION	DBO	CUSTOMERS	SALESORDERNUMBER		CustomerSalesReporting
16	CRMSystem	REFINED	DBO	CUSTOMERPI	SALESTERRITOR	SYSTEM1	CONSUMPTION	DBO	CUSTOMERC	SALESTERRITORYCOUNTRY		CustomerChurnReporting
17	CRMSystem	REFINED	DBO	CUSTOMERPI	SALESTERRITOR	SYSTEM1	CONSUMPTION	DBO	CUSTOMERC	SALESTERRITORYKEY		CustomerChurnReporting
18	CRMSystem	REFINED	DBO	CUSTOMERPI	SALESTERRITOR	SYSTEM1	CONSUMPTION	DBO	CUSTOMERC	SALESTERRITORYREGION		CustomerChurnReporting
19	CRMSystem	REFINED	DBO	CUSTOMERPI	TOTALPRODUCT	SYSTEM1	CONSUMPTION	DBO	PRODUCTSAL	TOTALPRODUCTCOST		ProductSalesReporting
20	CRMSystem	REFINED	DBO	CUSTOMERPI	UNITPRICE	SYSTEM1	CONSUMPTION	DBO	PRODUCTSAL	UNITPRICE		ProductSalesReporting
21	DEFAULT	RAW	DBO	##TEMP1	ADDRESSLINE1	DEFAULT	RAW	DBO	##TEMP2	FULLADDRESS	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
22	DEFAULT	RAW	DBO	##TEMP1	ADDRESSLINE2	DEFAULT	RAW	DBO	##TEMP2	ADDRESSLINE2	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
23	DEFAULT	RAW	DBO	##TEMP1	BIRTHDATE	DEFAULT	RAW	DBO	##TEMP2	BIRTHDATE	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
24	DEFAULT	RAW	DBO	##TEMP1	CITY	DEFAULT	RAW	DBO	##TEMP2	FULLADDRESS	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
25	DEFAULT	RAW	DBO	##TEMP1	COMMUTEDIST/	DEFAULT	RAW	DBO	##TEMP2	COMMUTEDIST	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
26	DEFAULT	RAW	DBO	##TEMP1	COUNTRYREGIC	DEFAULT	RAW	DBO	##TEMP2	COUNTRYREGI	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
27	DEFAULT	RAW	DBO	##TEMP1	CUSTOMERALTI	DEFAULT	RAW	DBO	##TEMP2	CUSTOMERALT	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
28	DEFAULT	RAW	DBO	##TEMP1	CUSTOMERKEY	DEFAULT	RAW	DBO	##TEMP2	CUSTOMERKEY	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone
29	DEFAULT	RAW	DBO	##TEMP1	DATEFIRSTPUR(	DEFAULT	RAW	DBO	##TEMP2	DATEFIRSTPUR	join_tables_anonymize_push_to_refined_zone	join_tables_anonymize_push_to_refined_zone

No	Column	Description
1	source_system	The name of the source system. Note This column is only shown when `useCollibraSystemName` is set to `true` in the lineage harvester configuration file.
2	source_database	The name of the source database.
3	source_schema	The name of the source schema.
4	source_table	The name of the source table.
5	source_column	The name of the source column.
6	target_system	The name of the target system. Note This column is only shown when `useCollibraSystemName` is set to `true` in the lineage harvester configuration file.

No	Column	Description
7	target_database	The name of the target database.
8	target_schema	The name of the target schema.
9	target_table	The name of the target table.
10	target_column	The name of the target column.
11	procedure_name	The name of the stored procedure. This column remains empty when an object in your technical lineage doesn't have stored procedure. Warning This column is deprecated and will be removed in the future.
12	query_name	The name of the specific source code fragment or transformation detail. You can use this name to search for more information in the **Sources** tab page.

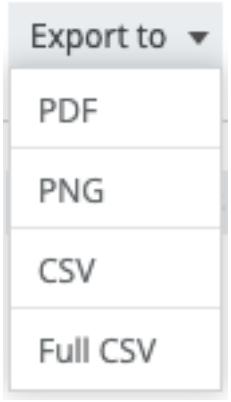
Tip The names of the source and target objects indicate the full path of the object. For example, the full name of a column is (system) > database > schema > table > column. This path is used to stitch your technical lineage objects to assets in Data Catalog.

Export the technical lineage information

If you want to share a technical lineage graph or the transformation logic of your technical lineage, for example with colleagues who don't have access to Collibra, you can export the information to PDF, PNG and CSV.

Steps

1. In the Technical lineage viewer, click the Settings tab.
2. Click **Export**.
3. Click the export type.



» The technical lineage information is downloaded.

Technical lineage troubleshooting

This section describes what you can do when you encounter issues running the lineage harvester, browsing through a technical lineage or stitching data source objects in your data source to existing assets in Data Catalog.

Technical lineage general troubleshooting

This topic contains the following information:

- Most common issues
- Testing connectivity
- Password errors

Most common issues

The following messages or other issues can appear when you run the lineage harvester, view a technical lineage or upload the new relations to Data Catalog via Collibra Data Lineage.

> Tip For a list of all error codes and messages that the lineage harvester displays, please see the lineage harvester error codes section.

Problem	Solution
You get the following error message: `Could not find or load main class lineage.lineage-harvester-<version nr.>`	This error message appears when the folder path to the lineage harvester is invalid. Check the folder path and make sure that it does not contain whitespaces.
You get the following error message: `Failed to load file '<file-name>'. If the file is not in UTF-8, please convert it accordingly.`	This error message appears if the lineage harvester tries to read a non-UTF-8 SQL file of a data source with connection type `SqlDirectory`. To solve this issue, convert all SQL files to UTF-8 and rerun the lineage harvester.
The lineage harvester does not connect to hosts using a proxy server.	Technical lineage does not support proxy server authentication, but you can connect to a proxy server. For complete details, including the necessary commands, see Connecting to a proxy server.

Problem	Solution
You get the following error message: `Source '<data source name> failed with exception: javax.net.ssl.SSLHandshakeException: General SSLEngine problem`	This message appears when the proxy server sends an unexpected certificate to the lineage harvester or when the default Java truststore is empty or outdated. First update Java and rerun the lineage harvester to see if that resolves the issue. If the same error message is shown, try the following: **On Windows** > Note In the following example commands, we refer to the techlin-gcp-us server. You should refer to the correct CollibraData Lineage server in the geographic location of your Collibra Data Intelligence Cloud environment. 1. Run the following command to extract the certificate from the Tableau server: `keytool -printcert -rfc -sslserver techlin-gcp-us.-collibra.com:443 > tableau-cer-t.crt` > Tip Replace the URL techlin-gcp-us.collibra.com with the URL for your Tableau server, which you specify in the lineage harvester configuration file. This will create a file called tableau-cert.crt in the folder where you run this command. 2. Run the following command to find the loc-

Problem	Solution

	ation of your JAVA_HOME:
	`echo %JAVA_HOME%`
	» The location path will be something like the following: `C:\Program Files\Java\jdk-17.0.2`
	3. Use the location path of your JAVA_HOME in the following command, to import the tableau-cert.crt file into the cacerts file found above.
	`keytool -importcert -file tableau-cert.crt -alias "Tableau-ProdServerCert" -keystore "C:\Program Files\Java\jdk-17.0.2\cacerts"`
	> Note You can specify as different alias, if you want.
	4. Run the following command:
	`keytool -list -keystore "C:\Program Files\Java\jdk-17.0.2\lib\security\cacerts" | findstr "Tableau"`
	5. Enter the keystore password.
	> Tip The password is typically `changeit`.
	» A list of all certificates that match the Tableau string in the "C:\Program Files\Java\jdk-17.0.2\cacerts" file is shown.

Problem	Solution

	> Tip In the list of certificates, look for the one that you imported in step 3. If it's listed, it means the "C:\Program Files\Java\jdk-17.0.2\cacerts" file has the certificate needed to validate the Tableau server.
	
	6. Run the following command to have the lineage harvester use the cacerts file that you just updated.
	`set JAVA_OPTS=-Djavax.net.ssl.trustStore="C:\Program Files\Java\jdk-17.0.2\lib\security\cacerts" -Djavax.net.ssl.trustStorePassword="changeit"`
	
	7. Run the following command to test the synchronization:
	`./lineage-harvester.bat full-sync -s tableau`
	
	On Linux

Problem	Solution

	Note
	• In the following example commands, we refer to the techlin-gcp-us server. You should refer to the correct CollibraData Lineage server in the geographic location of your Collibra Data Intelligence Cloud environment.
	• If you want to add an existing certificate to the Java Truststore, instead of creating a new Keystore, replace "<your keystore name>" in steps 2 and 3, with the path to the `cacerts` file in your Java installation, for example *%JAVA_HOME%\jre\lib\cacerts*.
	
	1. Use the following command to get a certificate from the corresponding techlin-gcp-us.com site, which is part of the CollibraData Lineage infrastructure:
	`openssl x509 -in <(openssl s_client -connect techlin-gcp-us.collibra.com:443 -prexit 2>/dev/null) -out techlin-gcp-us.crt`
	Tip If you already have a correctly formatted certificate on the server, you can skip this step.
	2. Add the certificate to the Java Truststore:
	`keytool -importcert -file techlin-gcp-us.crt -alias techlin-gcp-us -keystore <your keystore name> -storepass changeit`

Problem	Solution
	3. Run the lineage harvester and use the new truststore using the following parameter: `-Djavax.net.ssl.trustStore=<your keystore name>` Example To synchronize your data sources again, run the following command: `./bin/lineage-harvester full-sync -Djavax.net.ssl.trustStore=mykeystore`

You get the following error messages:
 	
In the lineage harvester log file:
`java.lang.Exception: No native library found for os.name=Linux, os.arch=x86_64, paths=[/org/sqlite/native/Linux/x86_64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]`
 	
In the console:
`Failed to load native library:<sqlite-file-name>. osinfo: Linux/x86_64 java.lang.UnsatisfiedLinkError: /tmp/<sqlite-file-name>: failed to map segment from shared object: Operation not permitted`
	The lineage harvester uses a temporary file containing an SQLite database as a cache file. That means that you need write permission to the `/tmp` folder.
	
	If this action failed, you can specify another directory with write permissions using `-Dorg.sqlite.tmpdir=<path to a temp directory>`.
	
	Example You have a temporary directory with write permissions. The path to this directory is `custom/temp`. Run the lineage harvester with the following command:
	`./bin/lineage-harvester -Dorg.sqlite.tmpdir=custom/temp full-sync`

Problem	Solution
You get the following error message: `Technical lineage is not enabled for this Catalog instance.`	First make sure that there are no spelling errors in the `Data Catalog` section of the configuration file. If your configuration file is configured correctly, but the issue is not solved, create a support ticket to enable Technical lineage for your Collibra Data Intelligence Cloud instance in Salesforce.
You get the following error message: `The size of the import file is too large (max size: 10 MB).`	The file you are trying to upload exceeds the size limit for uploaded files. Contact Collibra support to increase the maximum file size.
You get the following error message: `Source 'X' was never successfully processed..`	This message appears when a source that is specified in the lineage harvester configuration file has never been successfully processed by the Collibra Data Lineage server. You can either: • Remove source 'X' from the configuration file, and then run the command again. • Run a full-sync of source X, and then re-run the command that previously failed.
Technical lineage is unavailable because the selected table does not contain columns.	Technical lineage only includes tables that have columns. Add a relation of the type "Table contains/is part of Column" between your Table asset and Column assets.

Problem	Solution
You get the following message in your technical lineage: `The current asset doesn't have a technical lineage yet.`	This message appears if one or more of the following situations apply: • The data source of the current asset is not included in the configuration file. If you want a technical lineage for this asset, add its data source to the configuration file. • You have upgraded to the lineage harvester 1.3.0 or newer or you created a technical lineage for the first time. In this case, you may need to restart your DGC service before you can see the technical lineage. • You see parsing errors. For more information, see the Sources tab page. • The full name of one or more relevant assets does not match any of the names of the assets in the configuration file, which causes automatic stitching to fail. Make sure that the information in the configuration file and the Data Catalog physical data layer matches: ◦ The relevant assets have relations between each other, for example *Technology asset groups/is grouped by Technology asset> → <Database asset> contains/is part of <Schema asset> contains/is part of <Table asset> contains/is part of <Column asset>*. ◦ The full name of your System asset matches the name of your system or the name you used in the configuration

Problem	Solution
	file. ◦ The full name of your Database asset matches the name of your database or, for Google BigQuery your project, or the name you used in the configuration file. ◦ The full name of your Schema asset matches the name of the Schema of the data source or the name you used in the configuration file. Tip Make sure that the full path of each asset in Data Catalog matches the full path of the corresponding data object from your data source on the Stitching tab page.
You get one of the following messages: • `Nodes count exceeds the limit 350.` • `Edges count exceeds the limit 1000.`	This message appears when the technical lineage graph exceeds the limit of 350 nodes or 1,000 edges, and is too large to build. This happens, for example, if you have a table with many columns and you try to show the technical lineage of all columns in a table in one graph. If you see this message, we recommend that you browse through the technical lineage graph on the object level or select a single column in the Browse tab pane. Note You cannot manually change these limits.

Problem	Solution
You get the following error message in your technical lineage for a Microsoft SQL Server data source: "`Oops, no data flow founds in your SQL scripts. Make sure you upload DML queries like insert, update, merge that moves data between the tables.`"	This error message appears when you run the lineage harvester to create a technical lineage for a Microsoft SQL Server data source without having the correct permissions to the SQL Server. As a result, the lineage harvester processes empty files and there is no technical lineage available for this data source. Make sure you have at least the VIEW DEFINITION permission or sysadmin role in Microsoft SQL Server. > Note If you use multiple users, make sure that each one of them has the proper permissions.
The import job fails. > Note If the import job fails during import and the failing job is rolled back, you can have both old and new relations. The old relations were created during the first job and the new relations are created after the rollback. If more than one job is triggered, only the failed job is rolled back.	First, check the following: • The asset ID must exist. • The structure of the data must be correct. • The cardinality of relation types between asset types. Then, rerun the import of relations.

Problem	Solution
Relations are not changed as expected.	Check whether the lineage harvester refreshed the data source via a scheduled job. If the import job failed, then the data source was not refreshed and the previously created relations stay the same. If that happened, rerun the lineage harvester to import again.
Manual relations are overwritten.	We recommend that you do not manually add relations of the type "Data Element targets / sources Data Element" between asset types that are imported via the scheduled jobs. These relations are overwritten every time the scheduled job synchronizes the data source.
Ingesting Looker or Power BI assets fails.	For more information, see the following sections: • Looker troubleshooting. • Power BI troubleshooting

Testing connectivity

You can check whether the lineage harvester can connect to the Collibra Data Lineage server and Data Catalog.

1. Run the lineage harvester in command line.
2. Run the following command: `test-connection`.
   » The result shows if the lineage harvester can connect to the Collibra Data Lineage server and Data Catalog.

The logs will also show the IP addresses of the Collibra Data Lineage servers that you have to whitelist.

Password errors

If you mistyped the password or want to change an existing password, go to the lineage harvester folder > **config/pwd.conf** and delete the lines below. As a result, the lineage harvester will ask for the password again.

> Tip If you have the lineage harvester version 1.3.0 or newer, you can also provide your passwords via stdin or a password manager.

```
{
   "url" : "<URL>",
   "userName" : "<user>",
   "password" : "<password>"
}
```

Technical lineage known issues and limitations

The following table shows known issues and limitations in the current lineage harvester version.

> Important The success rate of a technical lineage, as shown in the Sources tab page, gives a good indication of the processing success. A success rate less than 100%, however, does not mean processing was unsuccessful. A parsing error, for example, which negatively affects the success rate, does not always negatively affect the completeness of the lineage.

Known issue	Description
Stitching results of BI source have a gray background.	Usually, data objects that Collibra Data Lineage stitches to assets in Data Catalog have a yellow background in the technical lineage graph. However, assets of BI sources, for example Power BI, that are stitched to other assets in Data Catalog currently have a gray background. This does not indicate that stitching failed. You can see which assets are stitched on the Stitching tab page.

Known issue	Description
The lineage harvester currently does not support Java version 16.	If you have Java version 16 and run the lineage harvester with the `full-sync` command, the harvester fails during the API key retrieval process. As a workaround, we recommend the following: 1. Set the JAVA_OPTS to the following: `JAVA_OPTS='--illegal-access=deny'` 2. Run the lineage harvester in the same command line window.
Collibra Data Lineage does not reuse the database model or DDL statements from other sources in the lineage harvester configuration file.	Currently, all sources in the lineage harvester configuration file are analyzed separately. As a result, the database model and DDL statements that are used for one source are not taken into account when analyzing another source. As a workaround, we recommend that you make sure that each source has all DDL statements that it needs to be processed properly. > Tip Saving the DDL statements in separate files and adding the preview "_" before their names might speed up the analysis of the DDL statements.
Harvesting an Amazon Redshift data source fails when using a CDATA JDBC driver.	If you use a CDATA JDBC driver to harvest metadata from an Amazon Redshift data source, you have to set the QueryPassthrough property in the connection configuration to `true`, otherwise the driver fails to execute the query.

Lineage harvester messages

A message code is shown in the lineage harvester logs when something goes wrong during the lineage harvester process. The message code indicates which part of the harvesting process was skipped or failed and provides steps to resolve it.

General lineage harvester messages

Message code	Description
MSG-LIN-1001	The current asset does not have a technical lineage yet. Only assets that are processed and stitched by Collibra Data Lineage have a Technical lineage. Look for the asset name in the navigation tree of the Browse tab pane, to see if the asset was processed. • If the asset name is not shown in the navigation tree, ensure that the data source of the asset is included in the configuration file. • If the asset name is shown in the navigation tree, ensure that you correctly prepared the Data Catalog physical data layer for technical lineage before you run the harvester. Specifically, the full path of each asset in Data Catalog must match the full path of the corresponding data object from your data source on the Stitching tab page. Less likely factors, such as your lineage harvester version and parsing errors can also lead to this error. For complete troubleshooting information, see Technical lineage general troubleshooting.
MSG-LIN-3000	This is an unknown or unclassified lineage harvester error. Create a support ticket to report the issue.

Message code	Description
MSG-LIN-3001	The lineage harvester was able to successfully connect to the Collibra Data Lineage servers, but received HTTP client error response. If the error message contains **Technical lineage is not enabled for this Catalog instance**, do the following: • Make sure that the URL to your Collibra Data Intelligence Cloud in the `catalog` section of the lineage harvester configuration file is correct. • Make sure that the username and password you use to sign in to Collibra are correct. • Make sure that Collibra Data Lineage is enabled for your Collibra environment. If the error message contains **Enter a valid URL**, do the following: • This error is caused by an invalid URL. Make sure that the URL to your Collibra Data Intelligence Cloud in the `catalog` section of the lineage harvester configuration file is correct. If the issue persists, please contact Collibra support or your customer success manager.
MSG-LIN-3002	The lineage harvester was able to successfully connect to the Collibra Data Lineage servers, but received an HTTP server error response. Wait a few minutes and then run the lineage harvester again. If the issue persists, please contact Collibra support or your customer success manager.

Message code	Description
MSG-LIN-3003	The lineage harvester failed to retrieve the API key of your Collibra Data Intelligence Cloud environment with Data Catalog from the Collibra Data Lineage servers due to network connectivity issues. To resolve this issue, do the following: • Check your network connectivity. • Make sure you have whitelisted the IP addresses of all Collibra Data Lineage servers. • Check your proxy settings. Tip You can test your connectivity using the `test-connectivity` command.
MSG-LIN-3004	Unable to determine the geographic location of your Collibra Data Intelligence Cloud environment. When you run the lineage harvester, it firsts connects to any available Collibra Data Lineage server to determine your cloud provider and geographic location of your Collibra environment. Then, the lineage harvester sends the harvested metadata to the Collibra Data Lineage sever with the same cloud provider and geographic location. In this case, the geographic location of your Collibra environment could not be determined. If the issue persists, please contact Collibra support or your customer success manager.

Message code	Description
MSG-LIN-4000	The Collibra Data Lineage server is unable to connect to Data Catalog. To resolve this issue, do the following: • Check your network connectivity. • Make sure that the URL to your Collibra Data Intelligence Cloud in the `catalog` section of the lineage harvester configuration file is correct. • Make sure the host names of all databases in the lineage harvester configuration file are correct. If the issue persists, please contact Collibra support or your customer success manager.

SQL scanner messages

Message code	Steps to resolve the issue
MSG-LIN-5001	This is an unexpected error. Create a support ticket to report your issue.

Upgrade the lineage harvester

Each new lineage harvester adds features and enhancements to the previous version. We highly recommend that you always use the newest lineage harvester available.

> Tip If you want to know the difference between versions of the lineage harvester, see the lineage harvester changelog.

Upgrade to lineage harvester 1.3.0 and newer

The lineage harvester 1.3.0 enables you to connect to a Collibra Data Lineage server, based on your geolocation and cloud provider.

You only have to follow this upgrade procedure when you upgrade from lineage harvester 1.2.1 or older to lineage harvester 1.3.0 or newer or any time the server's geolocation or cloud provider changes.

> Tip We highly recommend that you always use the newest lineage harvester.

Steps

1. If you have strict firewall rules, whitelist one of the following IP addresses, based on your Collibra geolocation and cloud provider:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)
2. Download lineage harvester 1.3.0 or newer, from the Collibra Downloads page.
3. Install the lineage harvester.
4. Migrate the data sources in your old configuration file to the configuration file in the new lineage harvester folder.
5. If your old configuration file had a `techlin` section, remove this section and all its properties.
6. Optionally, add or remove data sources in your lineage harvester configuration file.
7. Use the `full-sync` command to synchronize all data sources in your configuration file.
   » The lineage harvester uploads your data sources to the Collibra Data Lineage server with the new IP address.

> Note If you have previously ingested Power BI, you must run the Power BI harvester again before you run the lineage harvester.

What's next?

You can now access your technical lineage via a Column, Table, Power BI Column or Looker Look asset page.

Reuse a configuration file in a new lineage harvester

When you created a technical lineage using an older lineage harvester, you can easily upgrade to the newest lineage harvester and reuse your configuration file.

> Tip If you want to upgrade from a lineage harvester version older than 1.3.0 to a newer version, please follow the steps in the lineage harvester upgrade topic.

Steps

1. Download the newest lineage harvester from the Collibra Downloads page.
2. Install the lineage harvester.
3. Migrate the data sources in your old configuration file to the configuration file in the new lineage harvester folder.
4. Optionally, add or remove data sources in your lineage harvester configuration file.
5. Use the `full-sync` command to synchronize all data sources in your configuration file.
   » The lineage harvester synchronizes your data sources on the Collibra Data Lineage server and refreshes your technical lineage.

lineage harvester change log

Collibra Data Lineage is updated and improved on a regular basis. On this page, you can see the most important changes between different versions of the lineage harvester. For a complete list, see the release notes.

> Note In the documentation, we assume that you have the most recent version of the lineage harvester. We highly recommend to download and use the newest lineage harvester from the Collibra downloads page even if you are on an older version of Collibra Data Intelligence Cloud.

> Warning If you upgrade to lineage harvester 1.3.0 or newer, you have to follow an upgrade procedure.

The following list contains the most important changes to the lineage harvester and its configuration file.

Changed in version	New lineage harvester improvements
2022.04	• You can now use the databaseMapping property in your Tableau <source ID> configuration file, to map a Tableau technical database name to the real database name. • When providing connection definitions for Informatica PowerCenter, the dbname property is no longer case-sensitive. • Collibra Data Lineage now supports calculated fields for embedded data sources that are published. • The display name for Looker Data Set assets now uses the 'label' property, which provides an easier-to-read name. • Improvements related to integrating Informatica PowerCenter data sources: ◦ Collibra Data Lineage now correctly creates a technical lineage when useCollibraSystemName is set to true. ◦ Collibra Data Lineage now replaces parameters starting with a single "$" inside extracted queries. • Improvements related to integrating Informatica Intelligent Cloud Services data sources: ◦ The lineage harvester now supports InOut parameters for mapping tasks. The parameters are now loaded and their values are used to replace variables in custom SQL queries. ◦ Parameters (including those with numbers in their names) in SQL overrides are now correctly matched.

Changed in version	New lineage harvester improvements
2022.03	• By default, the lineage harvester no longer harvests images. If you want to include images, include the optional excludeImages property in your configuration file and set the value to false. • When ingesting Tableau metadata, you can now leave empty the collibraSystemName property in your configuration file, even if the useCollibraSystemName property is set to true. • The lineage harvester now correctly shows the help overview when you run the --help command. • Hive source now skips harvesting DDL of exclusively locked tables. • When you change the domain reference ID in the lineage harvester configuration file, Tableau assets are now successfully deleted from the previous domain and recreated in the new domain. • You no longer see a Fiber Failed error while running the lineage harvester. • Protobuf is upgraded to version 3.19.3. • Fixed an issue that was causing incomplete technical lineage and stitching issues when using custom SQL in Tableau. • Fixed an issue that resulted in a TableauHarvesterError when ingesting Tableau metadata via the linage harvester. • Fixed a NullPointerException when no column data type is harvested. • Fixed an issue that was causing the ingestion of Looker metadata to fail. • Fixed an issue that was causing a JsonParseError when ingesting Tableau metadata.
2022.02	
1.4.4	The lineage harvester now supports: • Technical lineage for Matillion. Redshift and Snowflake projects in Matillion are supported. • Snowflake syntax for the CONNECT BY clause.

Changed in version	New lineage harvester improvements
1.4.3	The lineage harvester log output now includes Collibra Data Lineage server processing information.
1.4.2	Collibra Data Lineage has improved Teradata parsing.
1.4.1	The lineage harvester for IBM DataStage now supports environment files.
1.4.1	You can now add connection information to the Informatica Intelligent Cloud Services <source ID> configuration file.
1.4.1	You can now request MicroStrategy as a lineage harvester integration in beta.
1.4.0	You can now request the following lineage harvester integrations in beta: • AWS Glue script annotations • Matillion • Power BI Report Server • SQL Server Reporting Services
1.4.0	The lineage harvester logs now shows message codes to inform you of an issue.
1.4.0	The user that runs the lineage harvester no longer need elevated permissions to access Snowflake metadata. You need a role that can access the snowflake shared read-only database. To access the shared database, the account administrator must grant IMPORTED PRIVILEGES on the shared database to the user that runs the lineage harvester.

Changed in version	New lineage harvester improvements
1.3.5	The lineage harvester configuration file and Power BI harvester configuration files now have a `useCollibraSystemName` property. You use this property to enable the harvesters to process the value in `collibraSystemName` properties and map the structure of the data source to system > database > schema > table > column, which you can see in the technical lineage Browse tab pane. By default, this property is set to `False`.
1.3.5	You can now create a separate configuration file for each data source to define the `collibraSystemName` property. For more information about this option, see the following topics: • The Informatica <source ID> configuration file • The IBM DataStage or SQL Server Integration Services connection definition configuration files. • The Informatica Intelligent Cloud Services <source ID> configuration file. • The Power BI <source ID> configuration file. • The Looker <source ID> configuration file. • The JSON files with a predefined lineage.
1.3.5	You can now use the `customConnectionProperties` field for Microsoft SQL Sever JDBC sources.
1.3.4	HiveQL sources no longer have connection type JDBC. You can only create a technical lineage for HiveQL sources via folder. Tip If you previously had a database section in the configuration file with a HiveQL source, change the section to match the properties of a directory section before you run the harvester 1.3.4 or newer.

Changed in version	New lineage harvester improvements
1.3.4	You can now create a technical lineage for Informatica Intelligent Cloud Services. Specifically, for the Cloud Data Integration service. You add the connection information to the lineage harvester configuration file.
1.3.4	You can now also add other data source connectors to the connection definition file for DataStage.
1.3.4	The lineage harvester configuration file for HiveQL, Spark SQL, PostgreSQL, Redshift and Snowflake data sources can now have a `customConnectionProperties` property to provide specific connection properties.
1.3.3	You can now use connection definitions to create a technical lineage for SQL Server Integration Services.
1.3.3	You can add multiple Google BigQuery projects in the configuration file in the "projectIDs" property. The "projectName" property is now deprecated.
1.3.2	Collibra Data Lineage can now process transformation logic for IBM DataStage.
1.3.2	The Collibra Data Lineage server now has an IP address for a server located in Canada, for Google cloud users: 35.197.182.41.
1.3.1	The Collibra Data Lineage server now has an IP address for a server located in Canada, for AWS users: 15.222.200.199.
1.3.1	You can now create a technical lineage for MySQL data sources.
1.3.0	The lineage harvester now gives you the option to provide your passwords via stdin or a password manager.
1.3.0	The lineage harvester now supports IBM InfoSphere DataStage.

Changed in version	New lineage harvester improvements
1.3.0	The lineage harvester now supports Looker integration.
1.3.0	The lineage harvester now connects to one of the servers with the following IP addresses: • 18.198.89.106 (techlin-aws-eu) • 54.242.194.190 (techlin-aws-us) • 35.205.146.124 (techlin-gcp-eu) • 34.73.33.120 (techlin-gcp-us) Note The lineage harvester connects to different servers based on your geographical location and cloud provider. If your location or cloud provider changes, the lineage harvester rescans all your data sources and you have to restart your DGC service.
1.2.1	You can now use the lineage harvester to import new Power BI assets, relations and a technical lineage into Data Catalog.
1.2.0	The `general` section of the configuration file shows the following: • A `catalog` section: This part contains the connection details needed to connect to Data Catalog. You no longer need an API key to connect to Collibra cloud. This part of the configuration file is optional and not shown when you create the file via the lineage harvester. You can no longer use it in lineage harvester 1.3.0. `{ "general": { "catalog" : { "url" : ""} },`
1.2.0	You can now create a technical lineage for Netezza and Sybase ASE data sources.

Changed in version	New lineage harvester improvements
1.2.0	Collibra Data Lineage now supports SSIS transformations.
1.1.7	You can now create a technical lineage for SQL Server Integration Services (SSIS).
1.1.7	You can now create a custom technical lineage using a JSON file.
1.1.3	You need to provide specific information necessary to connect to Collibra cloud in the `techlin` section of the configuration file. `{ "general": { "techlin": { "userKey": "my-userkey"},`
1.1.3	The `extractQueries` field is now removed from the configuration file. The queries of your database are downloaded automatically.
1.1.3	You can now create a technical lineage for Informatica PowerCenter.
1.1.1	You can now create a technical lineage for the following data sources: • Amazon Redshift • Azure SQL server • Google BigQuery • HiveQL • IBM DB2 • Microsoft SQL Server • Oracle • PostgreSQL • SAP Hana • Snowflake • Spark SQL • Teradata

Business Summary Lineage

The Business Summary Lineage is a representation of relations of the type "Data Element sources / targets Data Element" in a business diagram. It is not a separate diagram view, but refers to any diagram that contains that relation type. It allows you to trace data flows between registered databases and, as such, provides a summary of a technical lineage.

> Note Click here for an overview of the differences between Technical lineage and a diagram with Business Summary Lineage.

You can create a new diagram view including the Business Summary Lineage or you can select one of the existing diagram views that shows the relation "Data Element sources / targets Data Element" between Column assets of registered data sources and between BI assets and assets of registered data sources.

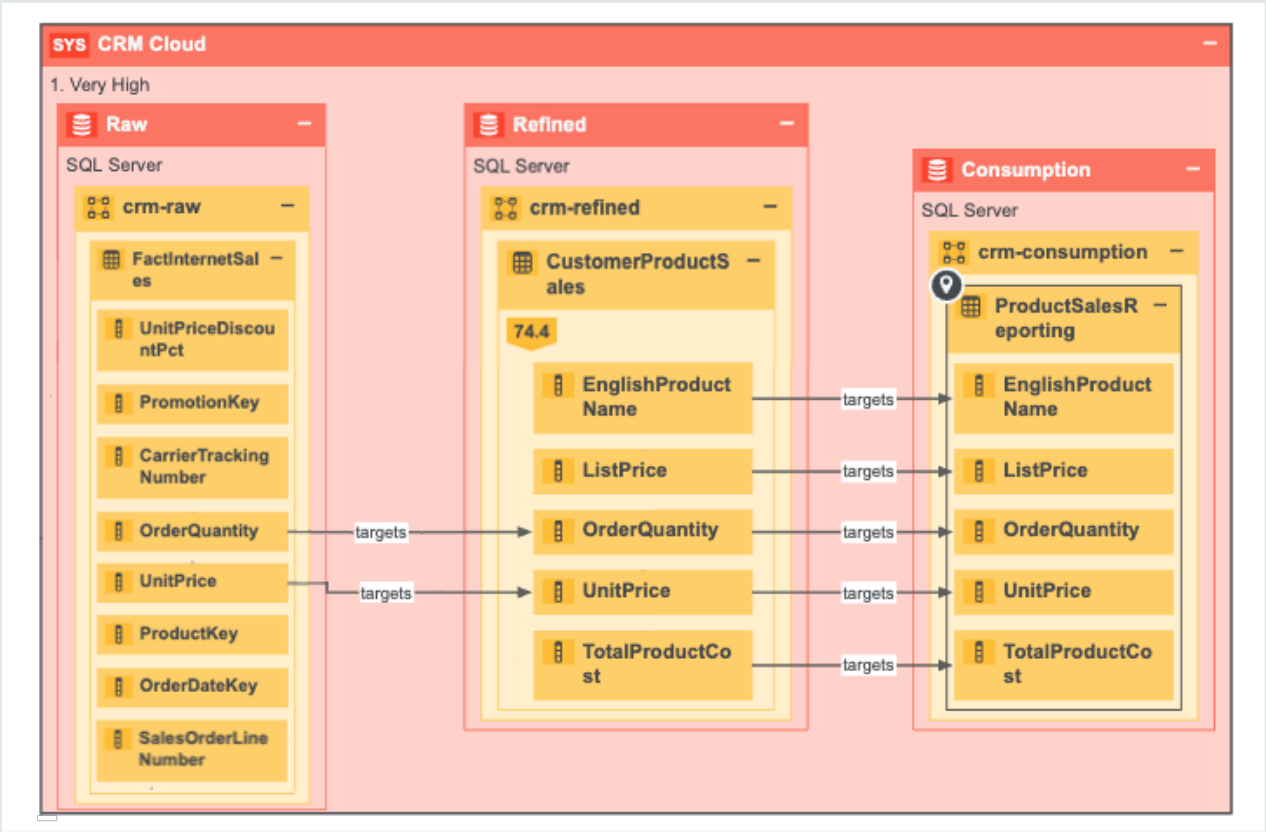
Before you can view a diagram with Business Summary Lineage, you have to:

- Register the data sources that you want to see in a diagram with Business Summary Lineage.
- Prepare a configuration file to create a technical lineage.
- Use the lineage harvester to upload the data sources in your configuration file to the Collibra Data Lineage server where they are scanned and processed.

Once the data sources are scanned, the Collibra Data Lineage server automatically pushes relations of the type "Data Element sources / targets Data Element" to Collibra Data Intelligence Cloud.

Example of a diagram with Business Summary Lineage

In this business diagram, you see that the Column assets of the Table asset CustomerProductSales have a relation of the type "Data Element sources / targets Data Element" to Column assets of other Table assets.



Differences between Technical lineage and diagrams with Business Summary Lineage

Technical lineage is a detailed lineage graph that shows where data objects are used and how they are transformed. A diagram with the Business Summary Lineage shows the relations between Data Assets in Data Catalog after stitching. Both map the flow of data, but a technical lineage provides a detailed overview of the data flow, while a diagram with Business Summary Lineage only provides a summary of it.

The Business Summary Lineage and a technical lineage are both visual representations of nodes. However, there are some key differences between them.

Business Summary Lineage	Technical lineage
A diagram with a Business Summary Lineage helps Business Analysts and other business users to understand their data by providing a summary of the technical lineage.	A technical lineage helps Data Engineers, Data Architects and similar personas to easily navigate to data objects in the data flows and find relevant source code fragments by providing a detailed lineage graph.
A diagram containing Business Summary Lineage is accessible via the Diagram tab pane of all assets.	A technical lineage is accessible via the tab pane of all Table assets and Column assets. You can view a technical lineage via the tab pane of Table assets and Column assets if you added their database as data sources in the configuration file.
A diagram shows assets and relations as defined in its diagram view. In the case of a Business Summary Lineage, the diagram shows, amongst others, relations of the type "Data Element targets / sources Data Element" between assets that exist in Data Catalog. Relations of this type are automatically created as part of the technical lineage process.	A technical lineage shows relations of the type "Data Element targets / sources Data Element" between all data objects in the data source. Relations of this type are automatically created as part of the technical lineage process. Note The data objects that you see in the technical lineage are: • Data Element assets for which you created the technical lineage, • Other objects, for example temporary tables and columns, that the lineage scanner collected from your data sources, but are not assets in Data Catalog.

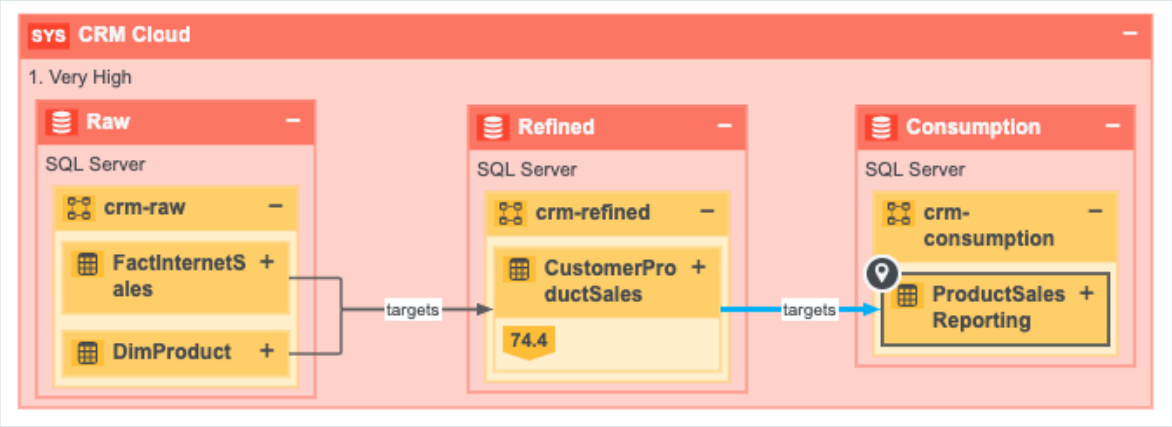
Business Summary Lineage	Technical lineage
A diagram with a Business Summary Lineage shows how registered data sources relate to each other.	Technical lineage shows how all data sources for which you create a technical lineage relate to each other. If the data source, or a part of the data source, is not registered in Data Catalog, the dependencies between the data elements in the data sources are still shown.

Example
You have created a technical lineage for four different databases:

- The first database, *Oracle*, is not ingested in Data Catalog and therefore has no assets in Data Catalog.
- The second database, *Raw*, contains tables that are ingested in Data Catalog, but also tables that are not ingested and therefore are not assets.
- The third and fourth database, *Refined* and *Consumption*, only contains data objects that are also assets in Data Catalog.

Technical lineage shows the data flow from all data objects in the first database, to the second, the third, and the fourth. Databases or data objects that are not ingested in Data Catalog and therefore are not assets, have a gray background.

A diagram with Business Summary Lineage only shows the relations between data objects that are also assets in Data Catalog, which means the data flow from assets in the second database to assets in the third, to assets in the fourth. The first database, which wasn't ingested, will not be shown on the diagram.



Dependencies

A dependency is a data object that is targeted by another data object. This is represented by a relation of the type "Data Element targets / sources Data Element", where the dependency is the tail.

There are two type of dependencies:

- a direct dependency: a data object that is the tail of a relation of the type "Data Element targets / sources Data Element".

  > Example If column A targets column B, then column B is the direct dependency of column A.

- an indirect dependency: a data object that is the target of a direct or another indirect dependency.

  > Example Column A targets column B, which on its turn targets column C. This means that column A indirectly targets column C, so column C is the indirect dependency of column A.

Working with Tableau

Tableau is business intelligence software that helps people see and understand their data. Integrating Tableau in Collibra Data Intelligence Cloud enables you to see metadata from Tableau Server and Tableau Online in CollibraData Catalog.

In this section, we describe how you can ingest Tableau metadata in CollibraData Catalog and synchronize the metadata using the lineage harvester, a standalone Java application. This method has numerous advantages over integration via the Data Catalog user interface.

> Important Please note the following important points regarding this integration method:
>
> - It is a cloud-only feature.
> - It is currently available only to customers who do not need to migrate existing Tableau assets to the new operating model, which applies only to the new integration method.
> - The new Tableau operating model is only available in Collibra versions 2021.10 and newer.
> - The two Tableau integration methods—Tableau integration via the Data Catalog and the new integration method via lineage harvester—coexist, and you are free to use the method of your choosing.
>
> We will soon make available a migration tool for those who would like to benefit from this integration method, but need to migrate existing Tableau assets.

Advantages and limitations of Tableau integration via lineage harvester

This section describes the advantages and limitations of using the lineage harvester to create Tableau assets and data in Data Catalog.

> Important Data Catalog uses Tableau's REST API to get metadata information and follows Tableau's requirements regarding authentication methods. As such, you need a Tableau user with access to the relevant Tableau sites. For more information, see the Tableau documentation.

Advantages

The following table shows the advantages of integrating Tableau metadata via the lineage harvester.

Advantage	Description
Credentials stored on-premises	Your Tableau credentials are no longer stored in the cloud. All credentials that you enter in the lineage harvester configuration file are encrypted and stored in the lineage harvester folder. You can also provide your passwords via command line to not store them anywhere.
Embedded data sources are eligible	You can ingest and synchronize embedded data sources and GZIP content, as well as published data sources.
Custom SQL parsing	This method uses our mature SQL scanners to parse custom SQL queries in Tableau reports and create detailed technical lineages.

Advantage	Description
Automatic stitching	The Collibra Data Lineage server that processes the Tableau metadata automatically stitches the data objects in Tableau to existing assets in Data Catalog. As a result, a relation of the type "Data Element sources / targets Data Element" is automatically created between Tableau Data Attribute assets and Column assets.
Technical lineage capabilities	A technical lineage is automatically created for Tableau assets. The technical lineage offers a visual representation of how data flows from one data object to another. Data Engineers, Data Architects and similar personas can benefit from such technical lineages, while Business Analysts and other business users can continue to benefit from diagrams with Business Summary Lineage.
Authentication via personal access tokens	You can authenticate with personal access tokens when using Tableau REST APIs, instead of user credentials.
Support for the Tableau Explorer role	You can use the Tableau Explorer role to import Tableau Data Attribute and Tableau Data Model assets and get lineage information. You no longer need a Tableau Admin role. > Note If you use the Explorer role, you also need the Data Management Add-on. For complete information, see the Tableau documentation.

Limitations

Currently, there are still a couple of limitations to integrating Tableau metadata via the lineage harvester:

- You ingest Tableau via the lineage harvester, instead of via the Data Catalog UI. All changes to the Tableau ingestion must be configured in the lineage harvester configuration file, instead of Data Catalog.
- We partially support Unions and Joins. For example, Unions created via the Tableau UI are not represented in Data Catalog. Tableau Data Sources created via custom SQL are supported.

Tableau terminology

The following table shows the Tableau terminology and corresponding asset types and terminology in Collibra Data Intelligence Cloud.

Tableau term	Description	Collibra equivalent
Site	A site is a stand-alone collection of content, such as projects, workbooks and users. Each site has its own URL and its own set of users.	Subcommunity and Tableau Site asset
Project	A project organizes related content resources. Content resources are workbooks, views and data sources.	Tableau Project asset
Workbook	A workbook is a collection of views.	Tableau Workbook asset
Dashboard	A dashboard is a collection of views from multiple worksheets.	Tableau Dashboard asset
Worksheet	A worksheet contains a single view, along with shelves, legends, and the Data pane.	Tableau Worksheet asset

Tableau term	Description	Collibra equivalent
Tableau data source	Tableau Data Sources consist of metadata that describe the connection information, information about how to access or refresh the data and customizations.	Tableau Data Model asset
Dimension	Dimensions contain qualitative values (such as names, dates, or geographical data).	Attribute type Role in Report on a Tableau Data Attribute asset page
Measure	Measures contain numeric, quantitative values that you can measure.	Attribute type Role in Report on a Tableau Data Attribute asset page
Tableau data attribute	Tableau Data Attributes define a property of a Tableau data entity.	Tableau Data Attribute asset
Tableau data entity	Tableau Data Entities are an abstraction of the physical implementation of database tables, used for Tableau report creation.	Tableau Data Model asset
Tableau data model	Tableau Data Models are an abstraction for the physical implementation of databases, schemas, files, etc., used for Tableau report creation.	Tableau Data Model asset
Tableau server	A Tableau server is a server on which Tableau users can publish data sources, as a means to share the data connections they've defined.	Tableau Server asset

Tableau asset types and domain types

The Tableau integration of Collibra Data Intelligence Cloud uses a specific subset of asset types and domain types. All of these come out of the box with your software.

The following table shows the asset and domain types that are used for the Tableau integration. Above each asset type you can see the parent asset types in the breadcrumbs.

Asset type	Description	Domain type
Business Asset ▸ Business Dimension ▸ BI Folder ▸ Tableau Project	Collection of Tableau workbooks and data sources.	BI Catalog
Business Asset ▸ Business Dimension ▸ BI Folder ▸ Tableau Site	Collection of content (workbooks, data sources, users, ...) that's walled off from any other content on that instance of Tableau Server.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Tableau View ▸ Tableau Dashboard	A collection of several worksheets and supporting information, shown on a single screen, so that you can simultaneously compare and monitor a variety of data.	BI Catalog

Asset type	Description	Domain type
Business Asset ▸ Report ▸ BI Report ▸ Tableau View ▸ Tableau Worksheet	A worksheet is a single sheet on which you can build views of your data.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Tableau Workbook	Collection of sheets. A sheet can be a worksheet, a dashboard or a story.	BI Catalog
Data Asset ▸ Data Element ▸ Data Attribute ▸ BI Data Attribute ▸ Tableau Data Attribute	A specification that defines a property of a Tableau data entity. Examples: CustomerBirthDate, EmployeeFirstName.	BI Catalog
Data Asset ▸ Data Structure ▸ Data Model ▸ BI Data Model ▸ Tableau Data Model	An abstraction from the physical implementation of database, schema, file, etc., used for Tableau report creation.	BI Catalog
Technology Asset ▸ Server ▸ BI Server ▸ Tableau Server	A visual analytics platform for creating interactive dashboards and rich visualisations	BI Catalog

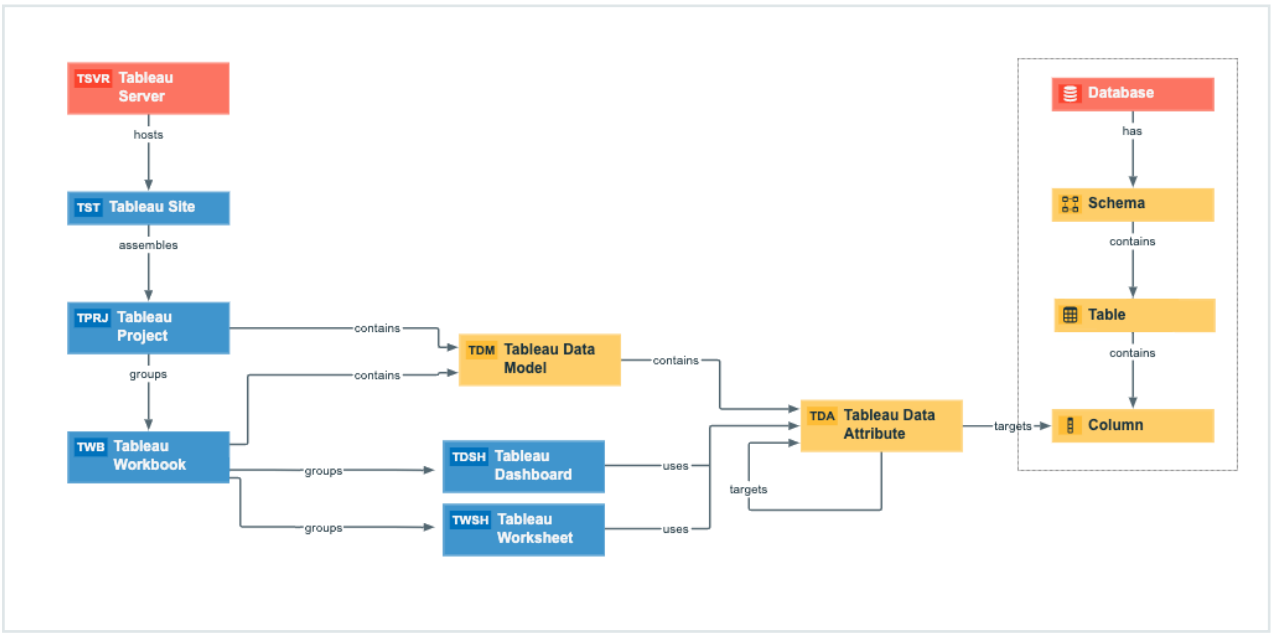
Tableau operating model

Synchronizing Tableau data means ingesting metadata from Tableau to your Collibra Data Intelligence Cloud environment. The metadata is represented as assets of specific types and their characteristics.

> Note
> - The assets have the same names as their counterparts in Tableau.
> - Some asset types are only created if the Tableau user has specific permissions.
> - Relations that were created between Tableau assets and other assets via a relation type in the Tableau operating model, are deleted upon synchronization. The same is true of any attribute types in the operating model that you add to Tableau assets. To ensure that the characteristics you add to Tableau assets are not deleted upon synchronization, be sure to use characteristics that are not part of the Tableau operating model.

Tableau operating model

The following image shows the relations between Tableau asset types.



Harvested metadata per asset type

This table shows the metadata for each Tableau asset type.

Asset type	Synchronized metadata
Tableau Server	• URL: The link to the data in Tableau • Description • Server hosts / is hosted in Business Dimension
Tableau Site	• URL: The link to the data in Tableau • Description • BI Folder assembles / Is assembled in BI Folder • Server hosts / is hosted in Business Dimension
Tableau Project	• Description • BI Folder assembles / is assembled in BI Folder • Business Dimension groups / is grouped into Report • Business Dimension source / is source of System
Tableau Work-book	• URL: The link to the data in Tableau • Description • Certified • Report Image • Document size • Document creation date • Document modification date • File size • Report groups / is grouped into Report • Tableau Workbook contains / contained in Tableau Data Model • Business Dimension groups / is grouped into Report

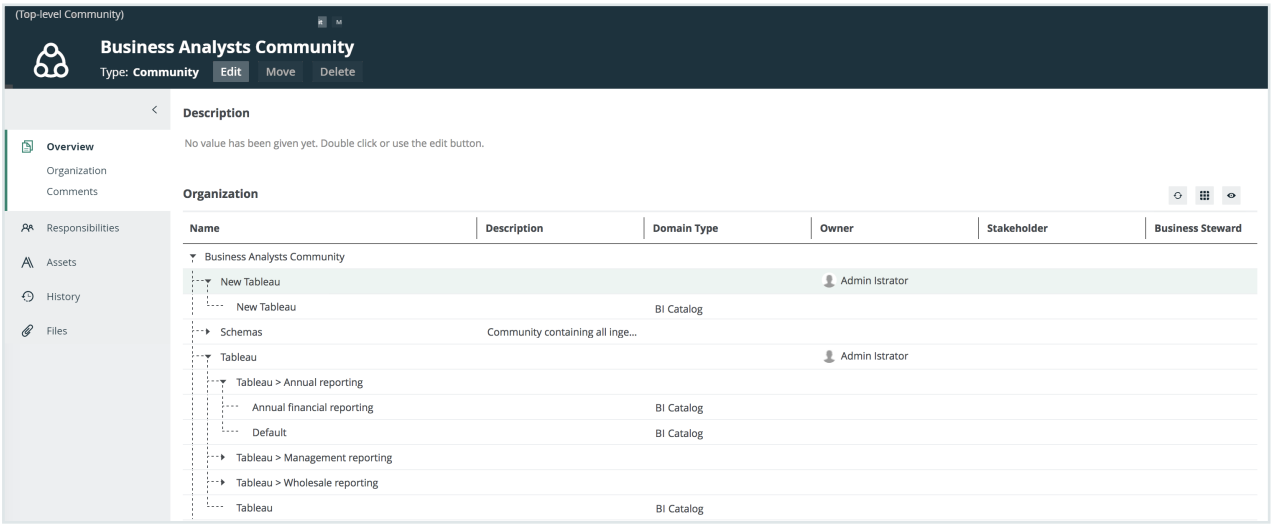
Asset type	Synchronized metadata
Tableau Dashboard	• URL: The link to the data in Tableau • Certified • Report image: The image of the report. > Note Images are downloaded and stored in Data Catalog. You can configure the maximum file size and content types of the Tableau images in the Collibra DGC service settings. • Document creation date • Document modification date • Visible on server • Report groups / is grouped into Report • Report uses / used in Report • Report uses / used in Data Attribute > Note Assets of this type are only created if the Tableau user has the Download/Save As permission on the workbook.

Asset type	Synchronized metadata
Tableau Worksheet	• URL: The link to the data in Tableau • Certified • Report image: The image of the report. Note Images are downloaded and stored in Data Catalog. You can configure the maximum file size and content types of the Tableau images in the Collibra DGC service settings. • Document creation date • Document modification date • Visible on server • Report groups / is grouped into Report • Report uses / used in Report • Report uses / used in Data Attribute Note Assets of this type are only created if the Tableau user has the Download/Save As permission on the workbook.
Tableau Data Attribute	• Data Type: The data type of a data asset, as it is declared by the data source. • Role in Report • Calculation Rule • Data Element targets / sources Data Element • Report uses / used in Data Attribute • BI Data Model contains / is part of BI Data Attribute Note Assets of this type are only created if the Tableau user has the Download/Save As permission on the data source.

Asset type	Synchronized metadata
Tableau Data Model	• Certified • Original Name: The name of the data source in Tableau • Document creation date • Document modification date • Business Dimension source / is source of System • BI Data Model contains / is part of BI Data Attribute • Tableau Workbook contains / contained in Tableau Data Model Note Assets of this type are only created if the Tableau user has the Download/Save As permission on the data source.

Example of ingested metadata

The following image shows an example structure after synchronizing Tableau.



Create a Tableau operating model diagram view

You can create a Tableau-specific diagram view, to visualize the operating model. The following procedure provides instruction on how to quickly create a new diagram view by copying and pasting the JSON code in the diagram view text editor.

Steps

1. Open an asset page.
2. In the tab pane, click **Diagram**.
   » The diagram appears in the default diagram view.
3. Click + to add a new view.
4. Click the **Text** tab, to switch to the diagram view text editor.
5. Click **Show me the JSON code** below this procedure, to expand the code.
6. Paste the code in diagram view text editor.
7. Click **Save**.
8. Edit the name and description of the diagram view, to suit your needs.

Show me the JSON code

```
{
   "nodes": [
        {
            "id": "Tableau Workbook",
            "type": {
            "id": "00000000-0000-0000-0000-110000000002"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Dashboard",
            "type": {
              "id": "00000000-0000-0000-0001-110000000301"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Worksheet",
            "type": {
              "id": "00000000-0000-0000-0001-110000000300"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Data Model",
            "type": {
              "id": "00000000-0000-0000-0000-110000000008"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Project",
```

```
            "type": {
              "id": "00000000-0000-0000-0000-110000000001"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Site",
            "type": {
              "id": "00000000-0000-0000-0000-110000000000"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Server",
            "type": {
              "id": "00000000-0000-0000-0000-110000000005"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Tableau Data Attribute",
            "type": {
              "id": "00000000-0000-0000-0000-110000000010"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Column",
            "type": {
              "id": "00000000-0000-0000-0000-000000031008"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Table",
            "type": {
              "id": "00000000-0000-0000-0000-000000031007"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Schema",
            "type": {
              "id": "00000000-0000-0000-0001-000400000002"
            },
            "layoutRegion": "context"
        },
        {
            "id": "Database",
            "type": {
```

```
            "id": "00000000-0000-0000-0000-000000031006"
          },
          "layoutRegion": "context"
      }
  ],
  "edges": [
      {
          "from": "Tableau Project",
          "to": "Tableau Workbook",
          "label": "",
          "style": "boxing",
          "type": {
            "id": "00000000-0000-0000-0000-120000000002"
          },
          "roleDirection": true
      },
      {
          "from": "Tableau Site",
          "to": "Tableau Project",
          "label": "",
          "style": "boxing",
          "type": {
            "id": "00000000-0000-0000-0000-120000000001"
          },
          "roleDirection": true
      },
      {
          "from": "Tableau Server",
          "to": "Tableau Site",
          "label": "",
          "style": "boxing",
          "type": {
            "id": "00000000-0000-0000-0000-120000000000"
          },
          "roleDirection": true
      },
      {
          "from": "Tableau Data Model",
          "to": "Tableau Data Attribute",
          "label": "",
          "style": "boxing",
          "type": {
            "id": "00000000-0000-0000-0000-000000007196"
          },
          "roleDirection": true
      },
      {
          "from": "Tableau Data Attribute",
          "to": "Tableau Data Attribute",
          "label": "",
```

```
            "style": "arrow",
            "type": {
              "id": "00000000-0000-0000-0000-000000007069"
            },
            "roleDirection": false
        },
        {
            "from": "Tableau Workbook",
            "to": "Tableau Data Model",
            "label": "",
            "style": "boxing",
            "type": {
              "id": "00000000-0000-0000-0000-120000000020"
            },
            "roleDirection": true
        },
        {
            "from": "Tableau Project",
            "to": "Tableau Data Model",
            "label": "",
            "style": "arrow",
            "type": {
              "id": "00000000-0000-0000-0000-120000000014"
            },
            "roleDirection": true
        },
        {
            "from": "Column",
            "to": "Column",
            "label": "",
            "style": "boxing",
            "type": {
              "id": "00000000-0000-0000-0000-000000007042"
            },
            "roleDirection": false
        },
        {
            "from": "Column",
            "to": "Table",
            "label": "",
            "style": "boxed",
            "type": {
              "id": "00000000-0000-0000-0000-000000007042"
            },
            "roleDirection": true
        },
        {
            "from": "Table",
            "to": "Schema",
            "label": "",
```

```
            "style": "boxed",
            "type": {
              "id": "00000000-0000-0000-0000-000000007043"
            },
            "roleDirection": false
        },
        {
            "from": "Schema",
            "to": "Database",
            "label": "",
            "style": "boxed",
            "type": {
              "id": "00000000-0000-0000-0000-000000007024"
            },
            "roleDirection": false
        },
        {
            "from": "Tableau Data Attribute",
            "to": "Tableau Worksheet",
            "label": "",
            "style": "arrow",
            "type": {
              "id": "00000000-0000-0000-0000-120000000021"
            },
            "roleDirection": false
        },
        {
            "from": "Tableau Workbook",
            "to": "Tableau Worksheet",
            "label": "",
            "style": "boxing",
            "type": {
              "id": "00000000-0000-0000-0000-120000000004"
            },
            "roleDirection": true
        },
        {
            "from": "Tableau Workbook",
            "to": "Tableau Dashboard",
            "label": "",
            "style": "boxing",
            "type": {
              "id": "00000000-0000-0000-0000-120000000004"
            },
            "roleDirection": true
        },
        {
            "from": "Tableau Worksheet",
            "to": "Tableau Dashboard",
            "label": "",
```

```
            "style": "arrow",
            "type": {
              "id": "00000000-0000-0000-0000-120000000007"
            },
            "roleDirection": false
        },
        {
            "from": "Tableau Data Attribute",
            "to": "Column",
            "label": "",
            "style": "arrow",
            "type": {
              "id": "00000000-0000-0000-0000-000000007069"
            },
            "roleDirection": false
        }
    ],
    "showOverview": false,
    "enableFilters": true,
    "showLabels": true,
    "showFields": true,
    "showLegend": true,
    "showPreview": true,
    "visitStrategy": "directed",
    "layout": "HierarchyLeftRight",
    "maxNodeLabelLength": 50,
    "maxEdgeLabelLength": 30,
    "layoutOptions": {
        "compactGroups": false,
        "componentArrangementPolicy": "topmost",
        "edgeBends": true,
        "edgeBundling": true,
        "edgeToEdgeDistance": 5,
        "minimumLayerDistance": "auto",
        "nodeToEdgeDistance": 5,
        "orthogonalRouting": true,
        "preciseNodeHeightCalculation": true,
        "recursiveGroupLayering": true,
        "separateLayers": true,
        "webWorkers": true,
        "nodePlacer": {
            "barycenterMode": true,
            "breakLongSegments": true,
            "groupCompactionStrategy": "none",
            "nodeCompaction": false,
            "straightenEdges": true
        }
    }
}
```

Supported data sources in Tableau

Tableau is business intelligence software that can integrate with various data sources. When you ingest Tableau metadata, Collibra Data Lineage tries to automatically stitch the metadata to data sources registered in Data Catalog. It also creates a Technical lineage that shows where metadata is used and how it transforms.

The following table shows the supported data sources in Tableau that have been tested, and whether or not technical lineage and stitching is supported for the data source.

We cannot guarantee that stitching works as expected for other data sources or versions.

> Tip For stitching, you must correctly prepare the Data Catalog physical data layer.

Data source	Version	Support for technical lineage	Support for stitching
Amazon Redshift	1.2.34.1058 and newer	Yes	Yes
Azure SQL server	Newest version	Yes	Yes
Azure SQL Data Warehouse	Newest version	Yes	Yes
Azure Synapse Analytics	Newest version	Yes	Yes
Dremio	20.0.0	Yes	Yes
Google BigQuery	Newest version	Yes	Yes
Greenplum	6.10 and newer	Yes	Yes
HiveQL (SQL-like statements)	2.3.5 and newer	Yes	Yes

Data source	Version	Support for technical lineage	Support for stitching
IBM DB2	11.5 and newer	Yes	Yes
Oracle	11g, 12c and newer	Yes	Yes
PostgreSQL	9.4, 9.5 and newer	Yes	Yes
Microsoft SQL Server	2014, 2016 and newer	Yes	Yes
MySQL	5.7, 8 and newer	Yes	Yes
Netezza	7.2.1.0 and newer	Yes	Yes
SAP Hana	2.00.40 and newer	Yes	Yes
Snowflake	Newest version	Yes	Yes
Spark SQL	2.4.3 and newer	Yes	Yes
Sybase Adaptive Server Enterprise	16.0 SP02 and newer	Yes	Yes
Teradata	15.0, 16.20.07.01 and newer	Yes	Yes

Automatic stitching

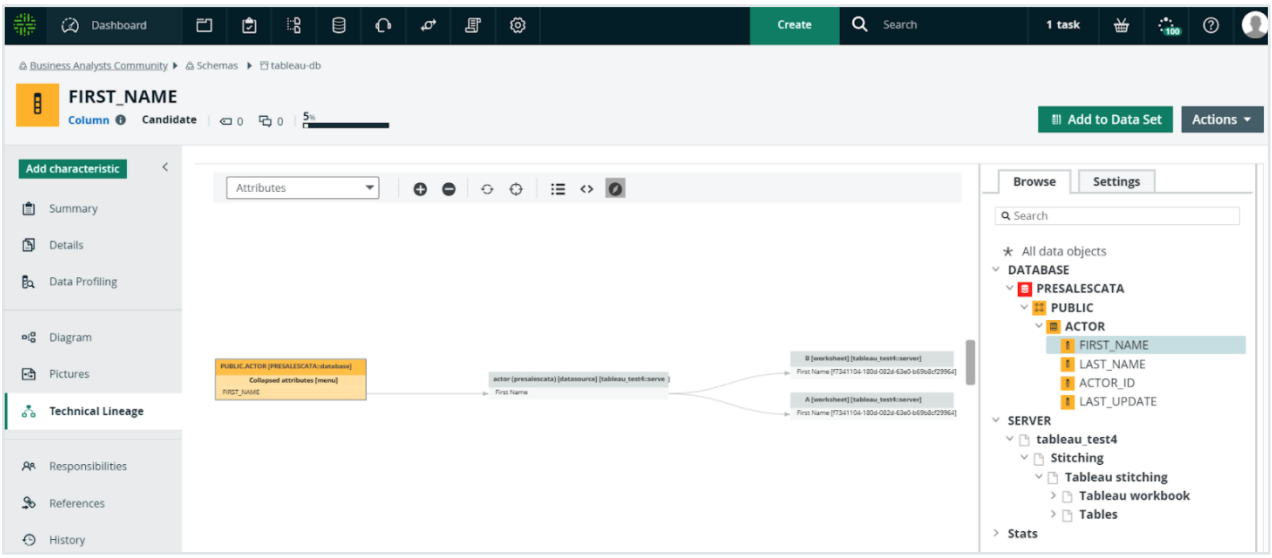
Stitching is a process that creates relations between database columns that are Column assets in Collibra Data Intelligence Cloud and BI assets representing the same database, specifically between:

- The assets that are created when you ingest Tableau.
- The assets that are created when you register a data source or import assets.

The lineage harvester harvests the Tableau source code and sends it to the Collibra Data Lineage server. The Collibra Data Lineage also collects the full names of assets ingested in Data Catalog and stitches them to data objects collected from Tableau. After processing the metadata, the Collibra Data Lineage server ingests the Tableau assets and their characteristics in Data Catalog. Tableau assets that are stitched now show a relation of the type "Data Element targets / sources Data Element" to the stitched asset. This relation type is also visualized in a technical lineage.

> Note If the Column asset is from a data source that is not supported by technical lineage, a standard SQL parser is used to try to visualize the column in a technical lineage, but the technical lineage might not be complete.

Usually, data objects that Collibra Data Lineage stitches to assets in Data Catalog have a yellow background in the technical lineage graph. However, the stitching results of BI sources currently have a gray background. This does not mean the stitching failed. You can see which assets are stitched in the Stitching tab page.



> Note When you ingest Tableau metadata, a technical lineage for Tableau Data Attribute assets is automatically created.

Stitching issues

To stitch assets in Data Catalog to data objects collected by the lineage harvester, the Collibra Data Lineage server looks at the full path of the assets in Data Catalog and the full path of Tableau assets. If the full paths match, the Collibra Data Lineage server automatically stitches them.

> Note Ensure that you correctly prepare the Data Catalog physical data layer.

The Technical lineage Stitching tab page shows the full paths of assets in Data Catalog and data objects collected from Tableau. To fix stitching issues, you can look up the full paths and make sure they match.

Technical lineage for Tableau

When you ingest Tableau metadata in Data Catalog, a technical lineage for Tableau Data Attribute assets is automatically created.

Permissions

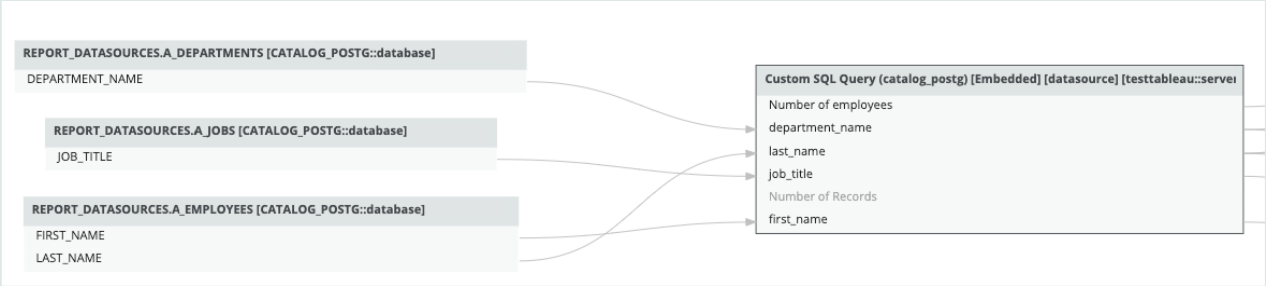
You can see the technical lineage of Tableau assets by clicking on the Technical lineage tab on the asset page of any Table or Column asset in Data Catalog when you have a Data Catalog global role with the Catalog and Technical lineage global permissions.

Technical lineage graph

The technical lineage graph shows relations of the type "Data Element sources / targets Data Element" between Tableau assets and other data objects in the data flow, for example between a Column asset and a Tableau Data Attribute asset. These relations are created during the Tableau ingestion process as a result of automatic stitching.

Example

The following technical lineage shows how data flows from a PostgreSQL data source to Tableau. It shows relations of the type "Data Element sources / targets Data Element" between the Column assets of the database and Tableau Data Attribute assets in Tableau. For example, Column asset *DEPARTMENT_NAME* has a relation of the type "Data Element sources / targets Data Element" to the Tableau Data Attribute asset *department_name*.



Sources tab page

The Sources tab page shows the transformation details that were analyzed and processed on the Collibra Data Lineage server and the results of this analysis. The success rate of the analysis indicates how complete the technical lineage is. There are a few limitations that prevent the Collibra Data Lineage server from processing all Tableau metadata.

> Important The Collibra Data Lineage server might not be able to process all complex Tableau metadata. This means that the success rate of a Tableau ingestion might not be 100%.

Overview Tableau integration steps

The Tableau integration enables you to harvest Tableau metadata and create new Tableau assets in Data Catalog. Collibra Data Intelligence Cloud analyzes and processes the metadata and presents it as specific asset types, retaining their original names.

Steps

The table below shows the steps and prerequisites required to integrate Tableau in Collibra via the lineage harvester.

Step	What?	Description	Prerequisites
1	Set up Tableau.	Before you start the Tableau integration in Data Catalog, make sure that the lineage harvester can reach the Tableau metadata. Perform these tasks before you start the actual Tableau ingestion process. Warning Because these tasks are performed outside of Collibra, it is possible that the content changes without us knowing. We strongly recommend that you carefully read the source documentation.	• You have a Tableau sub-scription.

Step	What?	Description	Prerequisites
3	Create a new domain.	Before you can ingest Tableau metadata, you have to create a new domain or choose an existing domain to store the new Tableau assets. Warning If you are using Collibra Data Intelligence Cloud 2021.11 or older, you have to add all Tableau attributes in the operating model to a scope and create a scoped assignment before you ingest Tableau via the lineage harvester. For complete information and step-by-step instruction, see Tableau general troubleshooting.	You have a resource role with the following resource permissions: • Domain: Add

Step	What?	Description	Prerequisites
4	Prepare the physical data layer.	You prepare Data Catalog's physical data layer to enable Data Catalog to automatically stitch the Tableau assets to existing assets in Data Catalog.	• You have a global role with the Catalog global permission, for example Catalog Author. • You have set up the JDBC driver of your source data, for example Snowflake. • You have a resource role with the following resource permissions on the **Schema** community: ◦ Asset > add ◦ Attribute > add ◦ Domain > add ◦ Attachment > add • You have the permissions to retrieve the metadata of the following database components through the JDBC Driver Database Metadata methods: ◦ Schemas ◦ Tables ◦ Columns

Step	What?	Description	Prerequisites
5	Download and install the lineage harvester	You use the lineage harvester to trigger the creation of Tableau assets, their relations and a technical lineage in Data Catalog. You can download the lineage harvester from the Collibra Product Resource Downloads page.	• Your environment meets the system requirements to install and use the lineage harvester.

Step	What?	Description	Prerequisites
6	Prepare the lineage harvester configuration file and run the lineage harvester.	You create a lineage harvester configuration file with Tableau connection information and run the lineage harvester to import the results of the Tableau integration and the technical lineage for Tableau into Data Catalog. As a result, Collibra creates new Tableau assets in Data Catalog and imports relations between these assets. It also creates a technical lineage for Tableau assets and other data sources in the lineage harvester configuration file.	• You have downloaded the lineage harvester version 2022.02 or newer. • Your environment meets the system requirements to install and run the lineage harvester. • You have a global role with the Catalog global permission, for example Catalog Author. • You have a global role with the Technical lineage global permission. • You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain: ◦ Asset: add ◦ Attribute: add ◦ Domain: add ◦ Attachment: add

Step	What?	Description	Prerequisites
7	View the Tableau assets and technical lineage	After the Tableau metadata is ingested in Data Catalog, you can go to the domain where you ingested Tableau and see the list of ingested Tableau assets. These assets are automatically stitched to existing assets in Data Catalog. You can also view the Tableau technical lineage. Warning When you run the lineage harvester, Collibra Data Lineage creates all Tableau assets in the BI Catalog domain (or domains) you specified. We highly recommend that you do not move these assets to other domains. If you move assets to other domains, they will be deleted and recreated in the initial BI Catalog domains when you synchronize Tableau. As a result, all manually added characteristics of those assets are lost.	You have a Data Catalog global role with the Catalog and Technical lineage global permissions.

Naming convention

When you synchronize Tableau, Collibra follows a strict naming convention for the names of the new assets. Each asset has a display name and full name. The full name represents the asset path from asset to the database it belongs to. You can freely edit the display name. However, you should never edit the full name, because Data Catalog may need it to

synchronize and stitch data sources. This may cause unexpected results and break the synchronization process.

> Warning We strongly recommend that you not edit the full names of any Tableau assets. Doing so will likely lead to errors during the synchronization process.

Set up Tableau

Before you ingest Tableau metadata in Data Catalog, you have to check if you have the right Tableau version, licenses, roles and permissions.

Tableau versions and licenses

Before you ingest Tableau metadata in Data Catalog via the lineage harvester, you must ensure that the lineage harvester can access and harvest the Tableau metadata.

> Important If you want to create a technical lineage and stitch your Tableau assets to assets in Data Catalog, you must enable the Tableau metadata API in Tableau.

Supported versions

- 2020.2
- 2020.3
- 2020.4
- 2021.1
- 2021.2
- 2021.3
- 2021.4

Tableau roles and permissions

The lineage harvester uses the Tableau Rest APIs and Tableau Metadata API to ingest the Tableau metadata. You need at least minimum permissions in Tableau to enable the lineage harvester to access the Tableau metadata and ingest it in Data Catalog.

Permissions on metadata

Permissions control who is allowed to see and manage external assets and which metadata (for both Tableau content and external assets) is shown through lineage.

> Note If Tableau Online or Tableau Server is not licensed with the Data Management Add-on, then by default, only admins can see database and table metadata through the Tableau Metadata API. You can turn on "derived permissions", to allow users to see metadata on external assets for the content that they own, or for the content that is published to a project for which they are a project leader or project owner. For complete information, see the Tableau documentation.

Minimum roles and permissions in Tableau

You need to following minimum roles and permissions to harvest Tableau metadata:

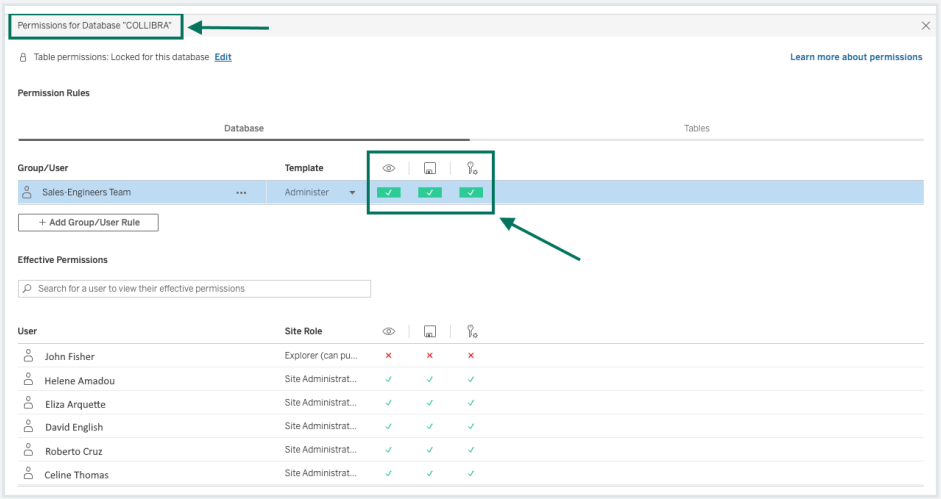
- You have a View permission on Tableau projects, workbooks and data sources you want to ingest.
- You have a Viewer or Explorer role with access to the Tableau REST API.

  > Note If you use the Explorer role, you also need the Data Management Add-on. This allows you to ingest metadata, but it does not allow for the creation of a technical lineage or stitching. For complete information, see:
  >
  > - Tableau ingestion results.
  > - The Tableau documentation.

Recommended roles and permissions in Tableau

For a full ingestion, we recommend the following roles and permissions in Tableau:

- You have at least a View permission on Tableau projects, workbooks and data sources you want to ingest.
- You have one of the following roles with access to the Tableau REST API:
  - Tableau Server Administrator or Tableau Site Administrator.
  - Explorer.

> Note If you use the Explorer role, you also need both:
> - The Data Management Add-on. For complete information, see the Tableau documentation.
> - Explicitly granted permissions, for every relevant database from which you want to ingest metadata.
>
> **See a screenshot of the required permissions granted at the database level**
>
> 
>

> Tip Tableau users with a Server Administrator role have access to the entire Tableau Server. Tableau users with a Site Administrator role can only be assigned to specific Tableau sites. As a result, if you have the Site Administrator role, only metadata from specific Tableau sites can be ingested in Data Catalog.

Tableau ingestion results

The following tables shows the ingestion results based on Tableau permissions.

By default, the lineage harvester uses both the Tableau REST API and the Tableau Metadata API, but you can limit the ingestion by allowing the lineage harvester to use only the Tableau REST API.

Tableau site role	Metadata API in Tableau	Result in Data Catalog
Viewer	Disabled	Tableau reports and data sources are ingested into Data Catalog, but with a limited scope. Resulting asset types: • Tableau Server • Tableau Site • Tableau Project • Tableau Data Model • Tableau Workbook • Tableau Worksheet Important We cannot retrieve lineage information or perform automatic stitching.
Viewer	Enabled	Tableau reports and data sources are ingested into Data Catalog, but with a limited scope. Resulting asset types: • Tableau Server • Tableau Site • Tableau Project • Tableau Data Model • Tableau Data Attribute • Tableau Workbook • Tableau Worksheet Important We cannot retrieve lineage information or perform automatic stitching.

Tableau site role	Metadata API in Tableau	Result in Data Catalog
Explorer, without the Data Management Add-on	Disabled	Tableau reports and data sources are ingested into Data Catalog, but with a limited scope. Resulting asset types: • Tableau Server • Tableau Site • Tableau Project • Tableau Dashboard • Tableau Data Model • Tableau Workbook • Tableau Worksheet Important We cannot retrieve lineage information or perform automatic stitching.
Explorer, without the Data Management Add-on	Enabled	Tableau reports and data sources are ingested into Data Catalog, but with a limited scope. Resulting asset types: • Tableau Server • Tableau Site • Tableau Project • Tableau Dashboard • Tableau Data Model • Tableau Data Attribute • Tableau Workbook • Tableau Worksheet Important We cannot retrieve lineage information or perform automatic stitching.

Tableau site role	Metadata API in Tableau	Result in Data Catalog
One of the following: • Tableau Server Administrator • Tableau Site Administrator • Explorer, with the Data Management Add-on	Disabled	Data Catalog creates new assets according to your content in Tableau using metadata in Tableau databases and tables. Resulting asset types: • Tableau Server • Tableau Site • Tableau Project • Tableau Data Model • Tableau Workbook • Tableau Dashboard • Tableau Worksheet Important We cannot retrieve lineage information or perform automatic stitching.

Tableau site role	Metadata API in Tableau	Result in Data Catalog
One of the following: • Tableau Server Administrator • Tableau Site Administrator • Explorer, with the Data Management Add-on	Enabled	Data Catalog creates new assets according to your content in Tableau using metadata in Tableau databases and tables. Resulting asset types: • Tableau Server • Tableau Site • Tableau Project • Tableau Data Model • Tableau Data Attribute • Tableau Workbook • Tableau Dashboard • Tableau Worksheet Note If Tableau Online or Tableau Server is not licensed with the Data Management Add-on, then by default, only admins can see database and table metadata through the Tableau Metadata API. You can turn on "derived permissions", to allow users to see metadata on external assets for the content that they own, or for the content that is published to a project for which they are a project leader or project owner. For complete information, see the Tableau documentation.

Prepare a domain for Tableau ingestion

During Tableau integration, Tableau assets are ingested in one or more specified domains in Collibra Data Intelligence Cloud. You then include the domain reference ID (or IDs) in the appropriate configuration file.

Prerequisites

- You have a resource role with the Domain > Add resource permission.

Steps

1. In the main menu, click the **Create** (+) button.
   » The **Create** dialog box appears.
2. Click the **Organization** tab.
3. Click a domain type from the list.
   If you clicked the wrong domain type here, you can change it in the **Type** field in the next screen.
   » The **Create Domain** dialog box appears.
4. Enter the required information.

Field	Description
Type	The domain type of the domain you are creating. In this case, you need to select *BI Catalog*.
Community	The community under which the domain will be located.
Name	The name of the new domain or domains. Tip You can create multiple domains in one go. To do this, press `Enter` after typing a value and then type the next. Domain names have to be unique in their parent community. If you type a name that already exists, it will appear in strike-through style.

5. Click **Create**.
6. Open your domain. If you created multiple domains, open each of them in turn.

7. Copy the reference ID of each domain you created.

   > Tip If you go to your domain, you can find the domain ID in the URL. The URL looks like: https://<yourcollibrainstance>/domain/**22258f64-40b6-4b16-9c08-c95f8ec0da26**?view=00000000-0000-0000-0000-000000040001. In this example, the domain ID is in bold.

8. Paste the domain reference ID (or IDs) in the appropriate configuration file, depending on whether you want to ingest Tableau assets in a single domain or multiple domains.
   For complete information on which properties and which configuration files to use, see the `domainId` property description in Prepare the lineage harvester configuration file for Tableau.

> Warning When you run the lineage harvester, Collibra Data Lineage creates all Tableau assets in the BI Catalog domain (or domains) you specified. We highly recommend that you do not move these assets to other domains. If you move assets to other domains, they will be deleted and recreated in the initial BI Catalog domains when you synchronize Tableau. As a result, all manually added characteristics of those assets are lost.

> Warning If you are using Collibra 2021.11 or older, you have to add all Tableau attributes in the operating model to a scope and create a scoped assignment before you ingest Tableau via the lineage harvester. For complete information and step-by-step instruction, see Tableau general troubleshooting.

Set up the lineage harvester for Tableau ingestion

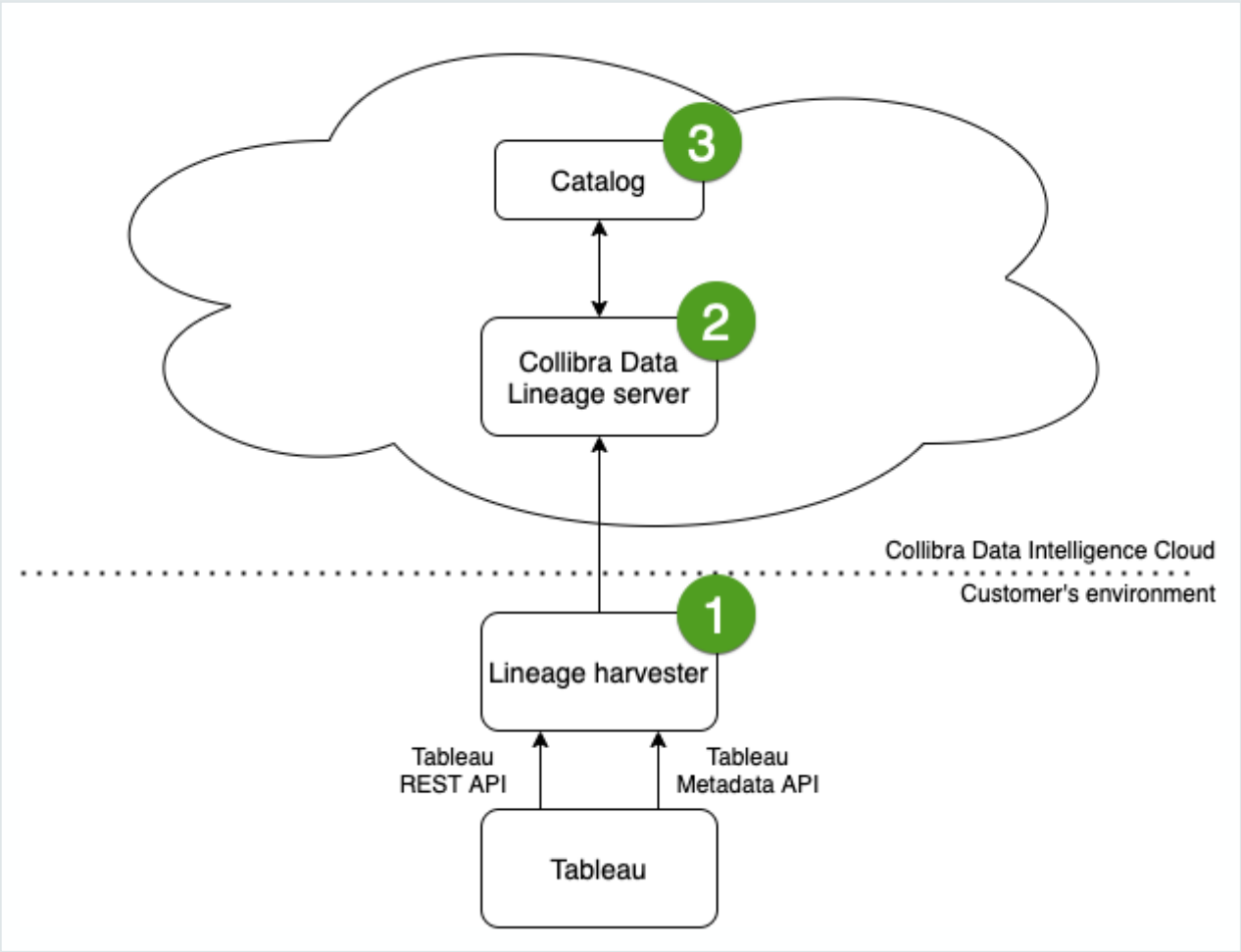
The lineage harvester is a software application that is required to collect your Tableau metadata and send it to the Collibra Data Lineage server, where the metadata is processed and new Tableau assets and relations are created.

> Note To ingest Tableau metadata into Data Catalog, you need lineage harvester 2022.02 or newer. We strongly recommend that you use the latest version of the lineage harvester.

Tableau ingestion workflow

You run the lineage harvester to start the Tableau ingestion workflow. When you initiate Tableau ingestion, each workflow component performs the following actions:

1. The lineage harvester:
    - Communicates with Tableau.
    - Harvests the Tableau metadata that will be ingested to Data Catalog.
    - Sends the Tableau metadata to the Collibra Data Lineage server.
2. The Collibra Data Lineage server:
    - Analyzes the Tableau metadata.
    - Creates new assets and relations.
    - Stitches existing assets in Data Catalog to Tableau assets.
    - Imports new Tableau assets and their relations in Data Catalog.
3. Data Catalog:
    - Shows new Tableau assets
    - Shows a Technical lineage for Tableau assets.
    - Shows stitching results between Tableau Data Attribute assets and Column assets.



Note This is the recommended workflow. If you do not want to use the Tableau Metadata API, you can disable it via the configuration file.

About the lineage harvester installation

You use the lineage harvester to collect source code from your data sources and create new relations between data elements from your data source and existing assets in Data Catalog.

The lineage harvester runs close to the data source and can harvest transformation logic like SQL scripts and ETL scripts from a specific location, for example a database table or a folder on a file system.

Note Collibra Data Lineage is a cloud-only feature.

Requirements

Type	Requirements
Software	Minimum requirements: • Java Runtime Environment version 11 or newer or OpenJDK 11 or newer. Recommended requirements: • Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.
Hardware	Minimum requirements: • 2 GB RAM • 1 GB free disk space Recommended requirements: • 4 GB RAM • 20 GB free disk space

Type	Requirements
Network	Firewall rules so that the lineage harvester can connect to: • The host names of all data sources in the lineage harvester configuration file. • All Collibra Data Lineage servers in your geographic location: ◦ 18.198.89.106 (techlin-aws-eu) ◦ 54.242.194.190 (techlin-aws-us) ◦ 15.222.200.199 (techlin-aws-ca) ◦ 35.205.146.124 (techlin-gcp-eu) ◦ 34.73.33.120 (techlin-gcp-us) ◦ 35.197.182.41 (techlin-gcp-au) ◦ 34.152.20.240 (techlin-gcp-ca) ◦ 51.105.241.132 (techlin-azure-eu) ◦ 20.102.44.39 (techlin-azure-us) Note The lineage harvester connects to different servers based on your geographic location and cloud provider. If your location or cloud provider changes, the lineage harvester rescans all your data sources. You have to whitelist all Collibra Data Lineage servers in your geographic location. In addition, we highly recommend that you always whitelist the techlin-aws-us server as a backup, in case the lineage harvester cannot connect to other Collibra Data Lineage servers.

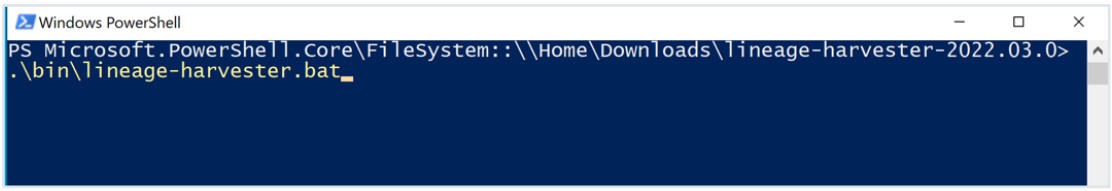
Note The lineage harvester uses port 443.

Installing the lineage harvester

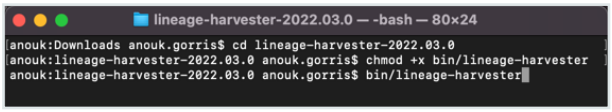
If you purchased Collibra Data Lineage, you can access the lineage harvester on the downloads page. To install the lineage harvester, do the following:

1. Download the lineage harvester.
2. Unzip the archive.
   » You can now access the lineage harvester folder.

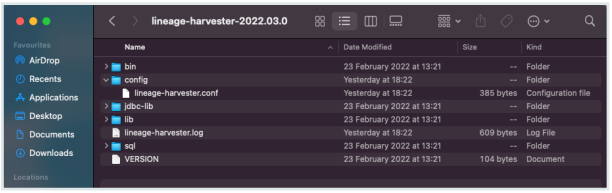
3. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



   » The lineage harvester is installed automatically. You can check the installation by running `./bin/lineage-harvester --help`.

> Note We highly recommend to always install and use the latest available lineage harvester.

Prepare the lineage harvester configuration file for Tableau

You have to prepare a configuration file before you run the lineage harvester. The lineage harvester collects your Tableau metadata and sends it to the Collibra Data Lineage server, where it is processed and analyzed. Collibra Data Intelligence Cloud then imports the Tableau assets and relations to Data Catalog.

Prerequisites

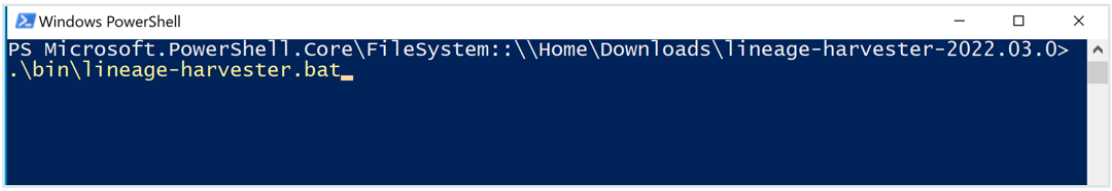
- You have Collibra Data Intelligence Cloud 2022.01 or newer.

  > Warning If you are using Collibra Data Intelligence Cloud 2021.11 or older, you have to add all Tableau attributes in the operating model to a scope and create a scoped assignment before you ingest Tableau via the lineage harvester. For complete information and step-by-step instruction, see Tableau general troubleshooting.

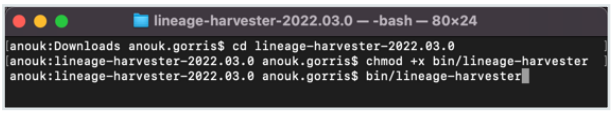
- You have the lineage harvester 2022.02 or newer.
- You have a global role that has the Manage all resources global permission.
- You have a global role with the Catalog global permission, for example Catalog Author.
- You have a global role with the Technical lineage global permission.
- You have created a BI Data Catalog domain in which you want to ingest the Tableau assets.
- You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain:
  - Asset: add
  - Attribute: add
  - Domain: add
  - Attachment: add
- You have downloaded the lineage harvester and you have the necessary system requirements to run it.
- You have tested your connectivity with the Tableau server.

Steps

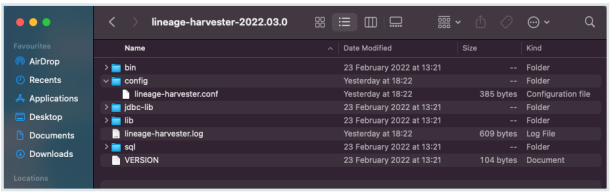
1. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



- For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



» An empty configuration file is created in the config folder.



2. Open the lineage-harvester.conf file and enter the values for each property.

Properties	Description
general	This section describes the connection information between the lineage harvester and Data Catalog.
catalog	This section contains information that is necessary to connect to Data Catalog.
url	The URL of your Collibra Data Intelligence Cloud environment. Note You can only enter the public URL of your Collibra DGC environment. Other URLs will not be accepted.
username	The username that you use to sign in to Collibra.

Properties	Description
useCollibraSystemName	Indication whether you want to use the system or server name of a data source to match to the System asset you created when you prepared the physical data layer. This is useful when you have multiple databases with the same name. By default, the useCollibraSystemName property is set to `false`. If you want to use it, set it to `true`. ◦ If you keep the property set to `false`, the lineage harvester ignores the collibraSystemName property in the rest of the configuration file. ◦ If you set the useCollibraSystemName property to `true`, the lineage harvester reads the value in the collibraSystemName property in all sections of the configuration file and in the Tableau <source ID> configuration file. Warning Unless you have multiple databases with the same name, we highly recommend that you keep the default value.
sources	This section contains all Tableau connection properties.
type	The kind of data source. In this case, the value has to be *Tableau*.

Properties	Description
id	The unique ID to identify the Tableau metadata that was uploaded to the Collibra Data Lineage. Tip This value can be anything as long as it is a unique. The lineage harvester uses the ID to identify a batch of data on the Collibra Data Lineage server.
url	The link to the data in Tableau.
username	The username you use to sign in to the Tableau server. Important If you want to use token-based authentication, you need to replace `username` with `tokenName`. You must specify either `username` or `tokenName`; if both exist, then `tokenName` is used.
tokenName	The lineage harvester authentication token. Note For token-based authentication, use this property in your lineage harvester configuration file, instead of the `username` property. If both properties are present, `tokenName` is used.

Properties	Description
siteIds	The site IDs of the Tableau sites that you want to include in the ingestion process. Warning Ensure that you specify the correct value. The correct value is the URL of the site to which you want to sign in. When you manually sign in to Tableau Server or Tableau Online, the site ID is the value that appears after **/site/** in the browser address bar. In the following example URLs, the site ID is `MarketingTeam`: ◦ Tableau Server: http://MyServer/#/site/MarketingTeam/projects ◦ Tableau Online: https://10ay.online.tableau.com/#/site/MarketingTeam/workbooks On Tableau Server, however, the URL of the Default site does not specify the site. For example, the URL for a view named Profits, on a site named Sales, is http://localhost/#/site/sales/views/profits. The URL for this same view on the Default site is http://localhost/#/views/profits. The site name Sales does not figure in the URL. If you can't see the site ID, leave this property empty: `siteIds: ""` Example If you want to ingest two Tableau sites "Site 1" and "Site 2", you can enter the following information in the siteIds property: ["site ID of Site 1", "site ID of Site 2"].

Properties	Description
siteNames	The site names of the corresponding site IDs. Important This property is: ◦ Optional for Tableau Server ◦ Mandatory for Tableau Online. Warning If you have Tableau Server and you don't use this property, you must delete it from your configuration file. Don't leave the property in the configuration file without a value.
restOnly	Indication whether or not you would like to use both the Tableau REST API and Tableau Metadata API to harvest Tableau metadata. ◦ `false` (default): The lineage harvester will use the REST API and Metadata API to harvest Tableau metadata. ◦ `true`: The lineage harvester will only use the REST API to harvest Tableau metadata. Warning If you only allow the lineage harvester to use the Tableau REST API, the harvester won't be able to process the necessary information for the technical lineage and the automatic stitching of Column assets to Tableau Data Attribute assets will not be possible.

Properties	Description
collibraSystemName	The name of the data source's system or server. You must include this property in your configuration file; however, you can leave it empty, even if the `useCollibraSystemName` property is set to `true`. If the `useCollibraSystemName` property is set to `true`, you must prepare a Tableau <source ID> configuration file to provide the system information.

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td>domainId</td><td>The unique reference ID of the domain in Collibra Data Intelligence Cloud in which you want to ingest the Tableau assets.

Tip You can ingest Tableau assets in one or more domains in Collibra. The following table identifies which properties and which configuration files to use, depending on whether you want to ingest in one or multiple domains.

<table>
<tr><th>If you want to...</th><th>Then...</th></tr>
<tr><td>Ingest in a single domain in Collibra</td><td>Refer to the single domain reference ID in this <code>domainID</code> property.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If you want to...</th><th>Then...</th></tr>
<tr><td>Ingest in multiple domains in Collibra</td><td>Do both of the following:
Mention a domain reference ID in this <code>domainID</code> property, for your Tableau Server asset.
Refer to all relevant domain reference IDs in the <code>domainMapping</code> section of the Tableau <source ID> configuration file, for your Tableau site, Tableau project and all child assets.
<blockquote>Important The <code>domainID</code> property represents the default domain. Tableau assets that are not mapped to specific domains via the <code>domainMapping</code> section of the Tableau <source ID> configuration file, for example Tableau Server assets, are ingested in this default domain.</blockquote></td></tr>
</table>
How do I find a domain reference ID?
Open the relevant domain in Collibra. The URL looks like: https://<yourcollibrainstance>/domain/22258f64-40b6-4b16-9c08-c95f8ec0da26?view=00000000-0000-0000-0000-000000040001. In this example, the reference ID is in bold.
</td></tr>
</table>

Properties	Description
excludeImages	Optional parameter for excluding the downloading of images. To exclude the downloading of images, set this property to `true`.

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td>paging</td><td>
Optional parameter for customizing the Tableau API pagination settings.
The default values are sufficient in most cases; however, you can decrease them to help mitigate node limit errors, or increase them to speed up API calls.

The complete list of pagination settings, descriptions and default values
<pre>
"paging": {
   "databasesPageSize": 100,
   "tablesPageSize": 100,
   "tablesColumnsPageSize": 100,
   "tableColumnsPageSize": 1000,
   "datasourcesPageSize": 50,
   "datasourcesFieldsPageSize": 50,
   "datasourceFieldsPageSize": 100,
   "worksheetsPageSize": 100,
   "worksheetsFieldsPageSize": 100,
   "worksheetFieldsPageSize": 1000,
   "dashboardsPageSize": 100,
   "columnsLimit": 20,
   "fieldsLimit": 20
   }
</pre>
Settings per metadata type and descriptions
<table>
<tr><th>Metadata type</th><th>Setting and description</th></tr>
<tr><td>Dashboard</td><td>◦ <code>dashboardsPageSize</code>: The number of dashboards per page.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Metadata type</th><th>Setting and description</th></tr>
<tr><td>Worksheet</td><td>◦ <code>worksheetsPageSize</code>: The number of worksheets per page.
◦ <code>worksheetsFieldsPageSize</code>: The number of worksheet fields per page.</td></tr>
<tr><td>Database</td><td>◦ <code>databasesPageSize</code>: The number of databases per page.</td></tr>
<tr><td>Table</td><td>◦ <code>tablesPageSize</code>: The number of tables per page.
◦ <code>tablesColumnsPageSize</code>: The number of table columns per page.</td></tr>
<tr><td>Table columns</td><td>◦ <code>tableColumnsPageSize</code>: The number of table columns per page.</td></tr>
<tr><td>Data source</td><td>◦ <code>datasourcesPageSize</code>: The number of data sources per page.
◦ <code>datasourcesFieldsPageSize</code>: The number of data source fields per page.
◦ <code>columnsLimit</code>: The number of data source field columns per page.
◦ <code>fieldsLimit</code> : The number of referenced data source fields per page.</td></tr>
</table>
</td></tr>
</table>

Properties	Description
	<table><tr><th>Metadata type</th><th>Setting and description</th></tr><tr><td>Data source field</td><td>◦ <code>datasourceFieldsPageSize</code>: The number of data source fields per page. ◦ <code>columnsLimit</code>: The number of data source field columns per page. ◦ <code>fieldsLimit</code> : The number of referenced data source fields per page.</td></tr></table>

3. Save the configuration file.
4. Start the lineage harvester again in the console and run the following command:
   - for Windows: `.\bin\lineage-harvester.bat full-sync`
   - for other operating systems: `./bin/lineage-harvester full-sync`
5. When prompted, enter the password or client secret to connect to your Collibra Data Intelligence Cloud and Tableau environment.
   » The passwords are encrypted and stored in **/config/pwd.conf**.

Example

```
{
 "general": {
   "catalog": {
     "url": "https://<organization>.collibra.com",
     "userName": "<your-collibra-username>",
     "useCollibraSystemName": false
   },
 },
 "sources": [
  {
   "type": "Tableau",
   "id": "unique-ID",
   "url": "URL to Tableau server",
   "username": "Admin",
   "siteIds": ["site ID of Tableau Site 1", "site ID of Tableau Site
2"],
```

```
   "siteNames": ["site name of Tableau Site 1", "site name of
Tableau Site 2"],
   "restOnly": false,
   "collibraSystemName": "tableau-system-name",
   "domainId": "Domain-resource-ID",
   "excludeImages": true,
   "paging": {
     "pagination-setting": 100,
     "pagination-setting-2": 100
   }
  }
 ]
}
```

What's next?

The lineage harvester triggers Collibra to import Tableau assets and their relations and create a technical lineage for Tableau Data Attribute assets.

If issues occur during the Tableau ingestion process, check the Tableau troubleshooting section to solve your problems.

To refresh the Tableau metadata, you can run the lineage harvester again or schedule jobs to run them automatically.

> Tip You can check the progress of the Tableau ingestion in Activities. The results field indicates how many relations were imported into Data Catalog.

Prepare the Tableau <source ID> configuration file

The lineage harvester uses the configuration file to connect to Tableau. However, you may need to provide additional information via a Tableau <source ID> configuration file. You use the Tableau <source ID> configuration file for the following reasons:

- To define your Tableau operating model.
- To provide additional information about databases and files in Tableau. For example, you can define the system name of databases in Tableau.

> Note This is only required when the `useCollibrasystemName` property in the lineage harvester configuration file is set to `true`.

Prerequisites

- The `useCollibraSystemName` in the lineage harvester configuration file is set to `true`.

Steps

1. Create a new JSON file in the lineage harvester **config** folder.
2. Give the JSON file the same name as the value of the `Id` property in the lineage harvester configuration file.

   > Example If the value of the `Id` property in the lineage harvester configuration file is `tableau-source-1`, then the name of your JSON file should be *tableau-source-1.conf*.

3. For each database in Tableau, add the following content to the JSON file:

Property	Description
collibraSystemNames	This section contains the system information for different Tableau data sources. Depending on the kind of data source or connection, you have to specify how to connect to this data source. Tip For more information, see the Tableau documentation. We also recommend to check the list of supported connectors in Tableau.

Property	Description
databases	This section contains connection information to one or more databases in Tableau. Tip If you do not have databases in Tableau, you can remove this section.
hostname	The host name of the database.
collibraSystemName	The system name of the database.
files	This section contains connection information to one or more files in Tableau. Tip If you do not have files in Tableau, you can remove this section.
filePath	The full path to the file. For example, the path to a JSON file.
collibraSystemName	The system name of the file.
connectors	This section contains connection information to one or more connectors in Tableau. Tip If you do not have connectors in Tableau, you can remove this section.
connectorUrl	The URL of the connector. For example, the URL to Google Analytics.
collibraSystemName	The system name of the connector.

Property	Description
cloudFiles	This section contains connection information to one or more cloud files in Tableau's input data. Tip If you do not have cloud files in Tableau, you can remove this section.
name	The name of the file. For example, the name of a Zendesk file.
collibraSystemName	The system name of the cloud file.

Property	Description
databaseMapping	The Tableau API returns a technical database name based on the hostname, instead of the actual database name, which breaks stitching. This property allows you to map a Tableau technical database name to the real database name, for example: `"databaseMapping": { "<hostname:port>":"<actual database name>" }`

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td></td><td>
Tip
The values that you specify for this property are not case sensitive. You can also use wildcards to capture multiple string combinations, for example:
<pre>"databaseMapping": {
  "it1166*":"IMMINT"
}</pre>
Show me the supported wildcards
<table>
<tr><th>Pattern</th><th>Description</th></tr>
<tr><td>*</td><td>Matches everything.</td></tr>
<tr><td>?</td><td>Matches any single character.</td></tr>
<tr><td>[seq]</td><td>Matches any character in "seq".</td></tr>
<tr><td>[!seq]</td><td>Matches any character not in "seq".</td></tr>
</table>
</td></tr>
</table>

Property	Description
domainMapping	This section defines in which domains in Collibra you want to ingest assets from your Tableau sites and Tableau projects. Important ◦ Use this property only if you want to ingest Tableau assets into multiple domains in Collibra Data Intelligence Cloud. If you want to ingest into a single domain, use only the `domainID` property in the lineage harvester configuration file. ◦ The `domainID` property in the lineage harvester configuration file represents the default domain. Tableau assets that are not mapped to specific domains via this `domainMapping` section, for example Tableau Server assets, are ingested in that default domain. Domain mapping is transitive, meaning that all resources, such as Tableau workbooks and data attributes in a parent Tableau site, project or sub-project, are ingested in the same domain as the parent. **Show me an example** Let's say that you have a Tableau site named "Site-1". You want to ingest all Tableau projects in "Site-1" in a domain named "Domain-1" in Collibra, with the exception of one Tableau project named "Project-Default", which you want to ingest in "Domain-2". You should configure the `domainMapping` section as follows.

Property	Description

	<pre>"domainMapping":
	{
	"Site-1": "reference-id-of-Domain-1",
	"Site-1 > Project-Default": "reference-id-of-Domain-2"
	}</pre>
	If you wanted to specify a domain for a sub-project of "Project-Default", you would use the `site name > project name > sub-project name` property, as described below.
	
	Tip For the properties in this `domainMapping` section, ensure that you maintain the spaces before and after ">", for example `"Site-1 > Project-Default"`. The spaces serve as a separator between the site and the projects.

site name	The unique reference ID of the domain in Collibra in which you want to ingest resources from the specified Tableau site. Important In the configuration file, use the actual site name, along with the domain reference ID, for example: `"Collibra_tab_partner_site": "afc8cfb0-91f1-4075-a3e5-7ce6d1f9bcc9"`

Property	Description
site name > project name	The unique reference ID of the domain in Collibra in which you want to ingest resources from the specified Tableau project. Important In the configuration file, use the actual site and project names, along with the domain reference ID, for example: `"Collibra_tab_partner_site > JB_Test_2812": "d224a1a5-43b4-43b2-8df0-ddf8f2726b82"`
site name > project name > sub-project name	The unique reference ID of the domain in Collibra in which you want to ingest resources from the specified Tableau sub-project. Important In the configuration file, use the actual site, project and sub-project names, along with the domain reference ID, for example: `"Collibra_tab_partner_site > JB_Test_2812 > ProjectJJ2": "d224a1a5-43b4-43b2-8df0-ddf8f2726b82"`

Property	Description
.	

```
{
  "collibraSystemNames": {
    "databases": [
      {
        "hostName": "tableau-server.us-east-
1.rds.amazonaws.com",
        "collibraSystemName": "public"
      }
    ],
    "files": [
      {
        "filePath": "C:\ProgramData\Tableau\Tableau
Server\data\files\sample.xls",
        "collibraSystemName": "sample-files"
      }
    ],
    "connectors": [
      {
        "connectorUrl": "tableau-server-connector-url.com",
        "collibraSystemName": "Oracle-connector"
      }
    ],
    "cloudFiles": [
      {
        "name": "file-name",
        "collibraSystemName": "FILE"
      }
    ]
  },
  "databaseMapping": {
    "it1166-imm-int.ccd4.eu-west-
1.rds.amazonaws.com:1521":"IMMINT"
  },
  "domainMapping": {
    "<site_name>": "domain-reference-id",
    "<site_name> > <project_name>": "domain-reference-id",
    "<site_name> > <project_name> > <subproject_name>":
"domain-reference-id"
   }
}
```

4. Save the <source ID> configuration file.

Schedule Tableau ingestion jobs

You can use Task Scheduler on Windows or Crontab on Mac and Linux to make the lineage harvester run scheduled jobs. In a scheduled job, the lineage harvester uploads the Tableau metadata information to Collibra.

Collibra automatically creates new Tableau assets and stitches the Tableau assets to existing data sources in Data Catalog at specific times, dates or intervals, using the information in your configuration file.

> Warning When you run the lineage harvester, Collibra Data Lineage creates all Tableau assets in the BI Catalog domain (or domains) you specified. We highly recommend that you do not move these assets to other domains. If you move assets to other domains, they will be deleted and recreated in the initial BI Catalog domains when you synchronize Tableau. As a result, all manually added characteristics of those assets are lost.

> Warning Relations that were manually created between Tableau assets and other assets via a relation type in the Tableau operating model, are deleted after a refresh of the Tableau metadata.

Tableau general troubleshooting

The following messages and issues can appear when you run the lineage harvester, view a technical lineage or upload the new relations to Data Catalog via Collibra Data Lineage.

Problem	Solution
You get connectivity issues with a 401001 error code. Unfortunately, 401001 is a very general error code, returned by a Tableau API, that can refer to many issues, including but not limited to the following: • The lineage harvester configuration file was configured with the wrong password or Tableau site ID. • SSO authentication was used, which is not supported.	Ensure that the user/token that you intend to use to ingest Tableau assets can authenticate to your Tableau APIs via the command line, from the server on which you intend to install and run the lineage harvester. You can test your ability to authenticate by making the signin API call, using a cURL command. You can also try checking the login request that the lineage harvester is sending to the Tableau server. For complete information and guidance on how to test your ability to connect to the Tableau server and authenticate, see Test connectivity with the Tableau server.
The lineage harvester does not connect to hosts using a proxy server.	Technical lineage does not support proxy server authentication, but you can connect to a proxy server. For complete details, including the necessary commands, see Connecting to a proxy server.

Problem	Solution

You get a TCP timeout error.	To avoid TCP timeout errors, try configuring the Linux TCP keepalive setting:
	1. Edit your /etc/sysctl.conf file:
	`# vi /etc/sysctl.conf`
	2. Add the following settings:
	`net.ipv4.tcp_keepalive_time = 60`
	`net.ipv4.tcp_keepalive_intvl = 10`
	`net.ipv4.tcp_keepalive_probes = 6`
	3. To load the settings, run the following command:
	`# sysctl -p`

Problem	Solution

You get the following error message:
`Source '<data source name> failed with exception: javax.net.ssl.SSLHandshakeException: General SSLEngine problem`
	This message appears when the proxy server sends an unexpected certificate to the lineage harvester or when the default Java truststore is empty or outdated.
	First update Java and rerun the lineage harvester to see if that resolves the issue. If the same error message is shown, try the following:
	On Windows
	> Note In the following example commands, we refer to the techlin-gcp-us server. You should refer to the correct CollibraData Lineage server in the geographic location of your Collibra Data Intelligence Cloud environment.
	1. Run the following command to extract the certificate from the Tableau server:
	`keytool -printcert -rfc -sslserver techlin-gcp-us.collibra.com:443 > tableau-cert.crt`
	> Tip Replace the URL techlin-gcp-us.collibra.com with the URL for your Tableau server, which you specify in the lineage harvester configuration file. This will create a file called tableau-cert.crt in the folder where you run this command.
	2. Run the following command to find the location of your JAVA_HOME:
	`echo %JAVA_HOME%`
	» The location path will be something like the following:
	`C:\Program Files\Java\jdk-17.0.2`
	3. Use the location path of your JAVA_HOME in the following command, to import the tableau-cert.crt file into the cacerts file found above.
	`keytool -importcert -file tableau-cert.crt -`

Problem	Solution

	`alias "TableauProdServerCert" -keystore "C:\Program Files\Java\jdk-17.0.2\cacerts"`
	> Note You can specify as different alias, if you want.
	4. Run the following command:
	`keytool -list -keystore "C:\Program Files\Java\jdk-17.0.2\lib\security\cacerts" | findstr "Tableau"`
	5. Enter the keystore password.
	> Tip The password is typically `changeit`.
	» A list of all certificates that match the Tableau string in the "C:\Program Files\Java\jdk-17.0.2\cacerts" file is shown.
	> Tip In the list of certificates, look for the one that you imported in step 3. If it's listed, it means the "C:\Program Files\Java\jdk-17.0.2\cacerts" file has the certificate needed to validate the Tableau server.
	6. Run the following command to have the lineage harvester use the cacerts file that you just updated.
	`set JAVA_OPTS=-Djavax.net.ssl.trustStore="C:\Program Files\Java\jdk-17.0.2\lib\security\cacerts" -Djavax.net.ssl.trustStorePassword="changeit"`
	7. Run the following command to test the synchronization:
	`./lineage-harvester.bat full-sync -s tableau`
	On Linux

Problem	Solution

	Note
	• In the following example commands, we refer to the techlin-gcp-us server. You should refer to the correct CollibraData Lineage server in the geographic location of your Collibra Data Intelligence Cloud environment.
	• If you want to add an existing certificate to the Java Truststore, instead of creating a new Keystore, replace "<your keystore name>" in steps 2 and 3, with the path to the `cacerts` file in your Java installation, for example *%JAVA_HOME%\jre\lib\cacerts*.
	
	1. Use the following command to get a certificate from the corresponding techlin-gcp-us.com site, which is part of the CollibraData Lineage infrastructure:
	`openssl x509 -in <(openssl s_client -connect techlin-gcp-us.collibra.com:443 -prexit 2>/dev/null) -out techlin-gcp-us.crt`
	Tip If you already have a correctly formatted certificate on the server, you can skip this step.
	2. Add the certificate to the Java Truststore:
	`keytool -importcert -file techlin-gcp-us.crt -alias techlin-gcp-us -keystore <your key-store name> -storepass changeit`
	3. Run the lineage harvester and use the new truststore using the following parameter:
	`-Djavax.net.ssl.trustStore=<your keystore name>`

Problem	Solution
	Example To synchronize your data sources again, run the following command: `./bin/lineage-harvester full-sync -Djavax.net.ssl.trustStore=mykeystore`

Problem	Solution
You get an external system ID mapping error	The error message looks similar to the following: `PROCESSING ERROR: "syncer.domain.DgcSyncError: java.lang.Exception: Unexpected DGC job status: ERROR Error message: { "type" : "MESSAGE", "message" : "A mapping for the external system id 'd0f3a21a2324fa117112409b-dea6ade7' and resource '59cf9293-fca1-4f78-99ab-31c150a23626' already exists." } Caused by: java.lang.Exception: Unexpected DGC job status: ERROR Error message: { "type" : "MESSAGE", "message" : "A mapping for the external system id 'd0f3a21a2324fa117112409b-dea6ade7' and resource '59cf9293-fca1-4f78-99ab-31c150a23626' already exists." }"` Please create a support ticket and provide your answers to the following two questions. Note Refer to the example error message above and replace the IDs of the external system and the mapped resource with those in the error message you received. • What is the asset type of the mapped asset? In this example, the asset with ID 59cf9293-fca1-4f78-99ab-31c150a23626? Tip To view the asset type, go to the following URL: <your-collibra-platform-url>/asset/59cf9293-fca1-4f78-99ab-31c150a23626 • What is the mapping definition?

Problem	Solution
	Tip To view the mapping definition, go to the following URL: <your-collibra-platform-url>/rest/2.0/mappings/externalSystem/d0f3a21a2324fa117112409bdea6ade7/mappedResource/59cf9293-fca1-4f78-99ab-31c150a23626

<table>
<tr><th>Problem</th><th>Solution</th></tr>
<tr><td>If you are using Collibra Data Intelligence Cloud 2021.11 or older, you have to add all Tableau attributes in the operating model to a scope and create a scoped assignment before you ingest Tableau via the lineage harvester.</td><td>

Show me how to add attributes to a scope and create a scoped assignment

Prerequisites

- You are using Collibra 2021.11 or older.
- You have a global role that has the System administration global permission.

Steps

1. In the main menu, click , then **Settings**.
   » The Collibra settings page opens.
2. In the tab pane, click **Scopes**.
3. Above the table, to the right, click **Add**.
   » The **Create Scope** dialog box appears.
4. Enter the required information.

Field	Description
Name	The name of the scope.
Description	The description of the scope, for example to add extra details.

5. Click **Save**.
6. Open a Tableau asset type:
   a. In the main menu, click , then **Settings**.
      » The Collibra settings page opens.
   b. In the tab pane, click **Asset Types**.
      » The asset type table appears.
   c. In the overview of asset types, click an asset type.
   » The **Asset type** editor opens.

</td></tr>
</table>

Problem	Solution
	Important You need to do this for each of the following asset types: ◦ Tableau Project ◦ Tableau Site ◦ Tableau Workbook ◦ Tableau Data Attribute ◦ Tableau Server There are other Tableau asset types, but they do not require the scoped assignment. 7. In the tab pane, click **Add assignment**. » The **Select scope for this assignment** dialog box appears. 8. Select the custom scope that you have created for Tableau assets. Note You can only add one scope at a time. 9. Click **Add assignment**. » The settings of the global assignment are copied into the selected scope. Warning After you've created the scoped assignment, do not change the assignment itself. The sole purpose of the scoped assignment is to ingest read-only attributes for which you normally need a system user. Note If you ingest Tableau metadata in a Collibra version 2021.09 or older, you must also manually create two new relation types and add them to the Tableau <source ID> configuration file.

Working with Power BI service

Power BI service is a cloud business intelligence software that helps you see and understand your data. You can ingest Power BI metadata in Data Catalog and create a technical lineage.

The Power BI service integration in Collibra Data Intelligence Cloud is not the same as the Power BI Report Server integration. If you want to ingest Power BI Report Server metadata in Collibra Data Intelligence Cloud, please read the Power BI Report Server section of the user guide.

> Note If you want to ingest Power BI metadata in Data Catalog, you have to purchase the Power BI connector and lineage feature.

Power BI terminology

Before you ingest Power BI, read more about the Power BI terminology and how it maps with the Collibra Data Intelligence Cloud asset types.

> Note For more information, see the Power BI documentation.

Power BI term	Description	Asset type in Collibra
Capacity	A resource that hosts Power BI Work-spaces.	Power BI Capacity
Dashboard	A collection of Power BI tiles with metrics from one or more Reports and Data Models.	Power BI Dashboard
Data Set	A collection of data that is used to create a Power BI report.	Power BI Data Model
Data Set Column	A column in a Power BI Data Model.	Power BI Column
Data Set Table	A table in a Power BI Data Model.	Power BI Table
Report	A detailed view of a Power BI Data Model, with visualizations of findings and insights.	Power BI Report

Power BI term	Description	Asset type in Collibra
Server or Tenant	A visual analytics platform for creating and storing Power BI Reports and Data Models.	Power BI Server
Tile	An element representing data on the Power BI Dashboard.	Power BI Tile
Workspace	A collection of Power BI Dashboards, Reports and Data Models.	Power BI Workspace

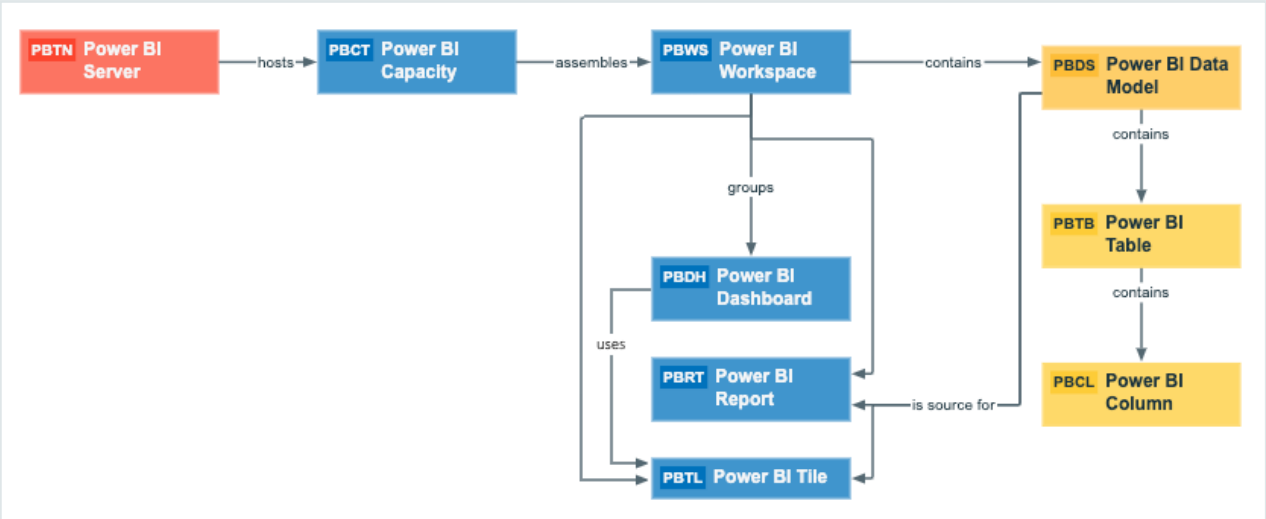
Power BI operating model

The Power BI harvester collects Power BI metadata and sends it to the Collibra Data Lineage server. Collibra processes the metadata and creates new Power BI assets and relations in Data Catalog. You can see them on the asset page overview or visualize them in a diagram or in a technical lineage.

> Note
> - The assets have the same names as their counterparts in Power BI. Full names and Display names cannot be changed in Data Catalog.
> - Depending on your Power BI subscription, it could be that not all asset types are created.
> - Asset types are only created if you have all specific Power BI and Data Catalog permissions.
> - All Power BI asset types are created in the same domain.
> - Relations that were manually created between Power BI assets and other assets in accordance with the relation types in the Power BI operating model, are deleted after a refresh of the Power BI metadata.

Power BI metadata overview

The following image shows the relations between Power BI asset types.



Harvested metadata per asset type

This table shows the harvested Power BI metadata for each Power BI asset type, assuming you have the necessary subscriptions and configurations for a full ingestion.

Asset type	Harvested Power BI metadata in Data Catalog
Power BI Capacity	• Full name • Display name • Server hosts / is hosted in Business Dimension • BI Folder assembles / is assembled in BI Folder
Power BI Column	• Full name • Display name • Description • Technical Data Type • Data Element targets / sources Data Element • BI Data Model contains / is part of BI Data Attribute • Data Entity contains / is part of Data Attribute

Asset type	Harvested Power BI metadata in Data Catalog
Power BI Dashboard	• Full name • Display name • Business Dimension groups / is grouped into Report • Report uses / is used in Report • Report related to / impacted by Business Asset
Power BI Data Model	• Full name • Display name • Data Asset is source for / source BI report • Data Entity is part of / contains Data Model • Data Model contains / is part of BI Data Attribute • BI Folder contains / contained in Data Asset
Power BI Report	• Full name • Display name • Business Dimension groups / is grouped into Report • Report related to / impacted by Business Asset • Report uses / is used in Report • BI Data Set is source for / source BI Report
Power BI Server	• Full name • Display name • Server hosts / is hosted in Business Dimension
Power BI Table	• Full name • Display name • Description • Data Entity contains / is part of Data Attribute • Data Entity is part of / contains Data Model

Asset type	Harvested Power BI metadata in Data Catalog
Power BI Tile	• Full name • Display name • Business Dimension groups / is grouped into Report • Report related to / impacted by Business Asset • Report uses / is used in Report • BI Data Set is source for / source BI Report
Power BI Workspace	• Full name • Display name • Description • BI Folder contains / contained in BI Data Model • BI Folder assembles / is assembled in BI Folder • Business Dimension groups / is grouped into Report

> Note The metadata that is shown on the assets' pages depends on the asset type's assignment. As a result, you might not see all harvested metadata on the asset's page by default.

Recommended hierarchy within a domain

You can enable hierarchies for the domain (or domains) in which your Power BI assets were ingested. Doing so makes it easier to understand the relation between your Power BI assets, when viewing the assets on the domain page.

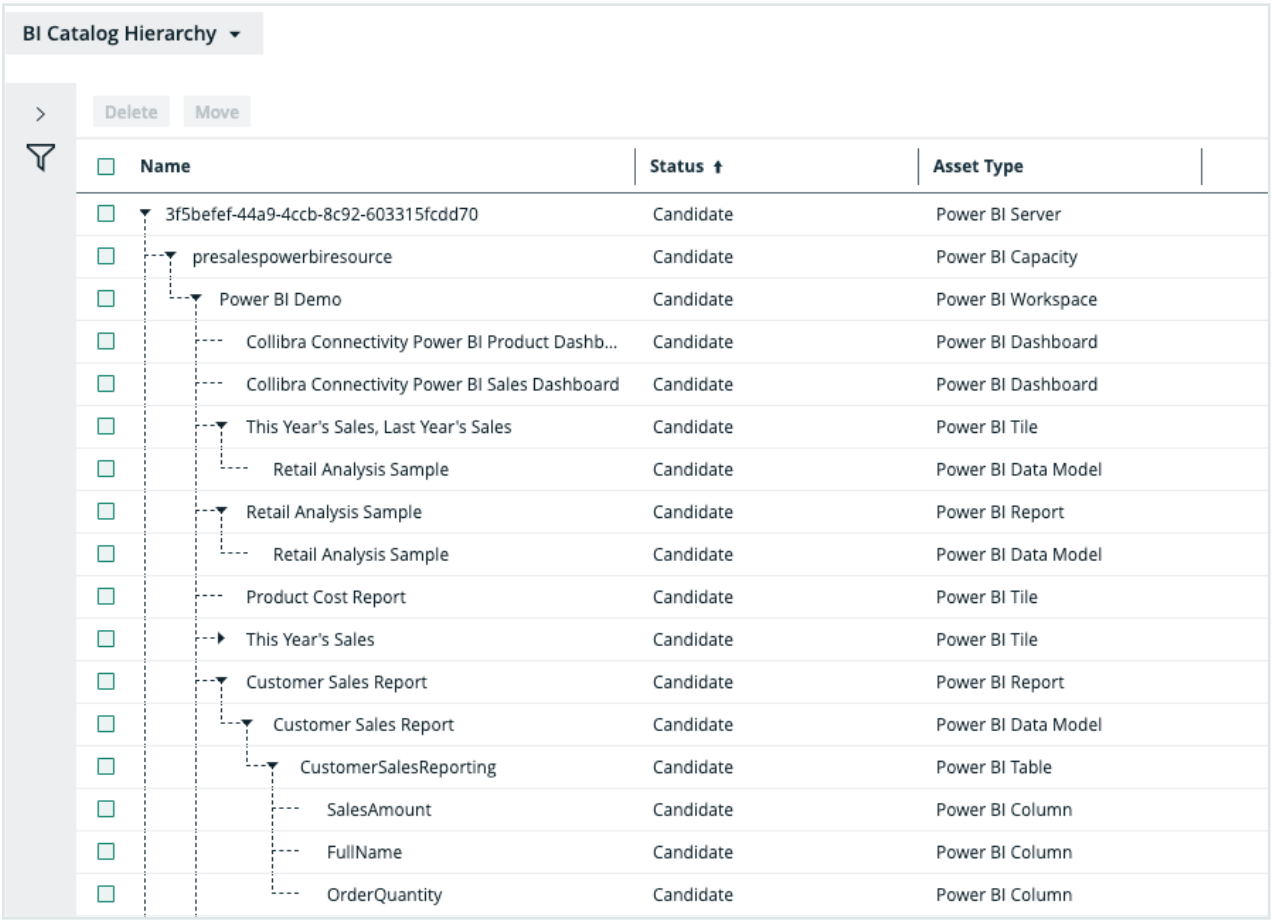
Follow these steps to enable and configure the recommended hierarchy.

Steps

1. Open the domain page of the relevant BI Catalog domain.
2. In the content toolbar, click .
   » The **Configure Hierarchy** dialog box appears.
3. Select **Enable Hierarchy**.
4. Select **Single path**.

5. Start typing and select each of the following relation types:
    - Server **hosts** Business Dimension
    - BI Folder **assembles** BI Folder
    - Business Dimension **groups** Report
    - BI Report **source** Data Asset
    - Data Model **contains** Data Entity
    - Data Entity **contains** Data Attribute
6. Click **Apply**.

The following image shows an example of a BI Catalog domain with hierarchies enabled.



> Note In an asset view, like the one shown in the previous image, if any asset is deleted, for example via synchronization or manual deletion, the view is recreated and the hierarchy is lost. In this case, you can again enable and configure the recommended hierarchy.

Power BI asset and domain types

The Power BI integration in Collibra Data Intelligence Cloud uses a specific subset of packaged asset types and domain types.

The following table contains the asset and domain types that are used for the Power BI integration. You can see the parent asset types in the breadcrumbs above each asset type.

Asset type	Description	Domain type
Business Asset ▸ Business Dimension ▸ BI Folder ▸ Power BI Capacity	A resource that hosts Power BI Workspaces.	BI Catalog
Business Asset ▸ Business Dimension ▸ BI Folder ▸ Power BI Workspace	A collection of Power BI Dashboards, Reports and Data Models.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Power BI Dashboard	A collection of Power BI tiles with metrics from one or more Reports and Data Models.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Power BI Report	A detailed view of a Power BI Data Model, with visualizations of findings and insights.	BI Catalog

Asset type	Description	Domain type
Business Asset ▸ Report ▸ BI Report ▸ Power BI Tile	An element representing data on the Power BI Dashboard.	BI Catalog
Data Asset ▸ Data Element ▸ Data Attribute ▸ BI Data Attribute ▸ Power BI Column	A column in a Power BI Data Model.	BI Catalog
Data Asset ▸ Data Structure ▸ Data Entity ▸ BI Data Entity ▸ Power BI Table	A table in a Power BI Data Model.	BI Catalog
Data Asset ▸ Data Structure ▸ Data Model ▸ BI Data Model ▸ Power BI Data Flow	A collection of tables that are created and managed in workspaces in the Power BI service.	BI Catalog
Data Asset ▸ Data Structure ▸ Data Model ▸ BI Data Model ▸ Power BI Data Model	A collection of data that is used to create a Power BI report.	BI Catalog

Asset type	Description	Domain type
Technology Asset ▸ Server ▸ BI Server ▸ Power BI Server	A visual analytics platform for creating and storing Power BI Reports and Data Models.	BI Catalog

Overview Power BI integration steps

The Power BI integration enables you to harvest Power BI metadata and create new Power BI assets in Data Catalog. Collibra analyzes and processes the BI metadata and presents it as specific asset types, retaining their original names.

> Tip To ingest Power BI metadata in Data Catalog, you need to run two different harvesters: the Power BI harvester and the lineage harvester. The order in which you run the harvesters is important. You first have to run the Power BI harvester to collect the metadata from your Power BI application and then run the lineage harvester to import new Power BI assets and their relations in Data Catalog. The Power BI ingestion workflow explains which roles the harvesters play in the Power BI ingestion process.

Steps

The table below shows the steps and prerequisites required to integrate Power BI in Data Catalog. These steps are best practices, which means that some of them might be optional, but highly recommended.

Step	What?	Description	Prerequisites
1	Set up a Power BI application.	Before you start the Power BI integration in Data Catalog, make sure that the Power BI harvester can reach the Power BI metadata. Perform these tasks before you start the actual Power BI ingestion process: • The authentication process. • The registration of your Power BI application in Microsoft Azure. • The Power BI roles and dedicated capacities for Power BI workspaces. • The required Power BI subscription. Warning Because these tasks are performed outside of Collibra, it is possible that the content changes without us knowing. We strongly recommend that you carefully read the source documentation.	You have a Power BI subscription.
2	Create a new domain.	Before you can ingest Power BI metadata, you have to create a new domain or choose an existing domain to store the new Power BI assets.	You have a resource role with the following resource permissions: • Domain: Add

Step	What?	Description	Prerequisites
3	Optionally, assign the attribute type State to the global assignment of the Power BI Workspace asset type	On Power BI Workspace asset pages, you can include the attribute type State, to show the state of ingested Power BI workspaces. To do so, you have to edit the global assignment of the Power BI Workspace asset type and assign the attribute type State. If you delete a Power BI workspace, the workspace is maintained for a 90-day grace period. During the grace period, the workspace has the state Deleted. When you ingest Power BI metadata in Data Catalog, this deleted workspace is ingested. For complete information on Power BI workspaces and possible states, see the Microsoft Power BI documentation.	You have a global role that has the System administration global permission.

Step	What?	Description	Prerequisites
4	Ingest or import assets from supported JDBC data sources.	The Collibra Data Lineage server connects to Data Catalog and reads the full paths of existing assets. When the full path matches the full path of assets in Power BI, the Collibra Data Lineage server automatically stitches them. Note Usually, data objects that Collibra Data Lineage stitches to assets in Data Catalog have a yellow background in the technical lineage graph. However, the stitching results of BI sources, for example Power BI, currently have a gray background. This does not indicate that stitching failed. You can see which assets are stitched in the Stitching tab page.	Permissions depend on how you ingest or import the assets.

Step	What?	Description	Prerequisites

5	Download and install the Power BI harvester.	You use the Power BI harvester to collect metadata from Power BI and upload it to Collibra where the metadata is scanned, processed and analyzed.
		
		You can download the Power BI harvester from the Collibra Product Resource Downloads page. The installer file contains the following:
		
		• a config folder with an empty configuration file.
		• a bin folder.
		• a TXT file with more information about the configuration file.
		• a BAT file that you use to run the harvester.
			• You have Collibra Data Intelligence Cloud 2020.11 or newer.
			• You have access to the Power BI harvester on the Downloads page.
			• Your environment meets the system requirements to install and use the Power BI harvester.
			• You have added Firewall rules so that the Power BI harvester can connect to the Collibra Data Lineage server with one of the following IP addresses:
			  • 18.198.89.106 (techlin-aws-eu)
			  • 54.242.194.190 (techlin-aws-us)
			  • 15.222.200.199 (techlin-aws-ca)
			  • 35.205.146.124 (techlin-gcp-eu)
			  • 34.73.33.120 (techlin-gcp-us)
			  • 35.197.182.41 (techlin-gcp-au)
			  • 34.152.20.240

Step	What?	Description	Prerequisites
			(techlin-gcp-ca) • 51.105.241.132 (techlin-azure-eu) • 20.102.44.39 (techlin-azure-us) Important Ingestion results vary according to your Power BI subscription.

Step	What?	Description	Prerequisites
6	Prepare the Power BI configuration file and run the Power BI harvester.	You create a configuration file to provide the connection information that you need to connect your Power BI application to Collibra and to the domain in which you want to ingest the Power BI assets. You can access an empty configuration file in the Power BI harvester installation folder. When you have created and saved the configuration file, you can run the Power BI harvester which uploads the Power BI metadata to Collibra.	• You have access to the Power BI harvester on the Downloads page. • You have completed all prerequisite tasks. • You have a dedicated domain to ingest the Power BI assets. • You have a global role with the Catalog global permission, for example Catalog Author. • You have a global role with the Technical lineage global permission. • You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain: ◦ Asset: add ◦ Attribute: add ◦ Domain: add ◦ Attachment: add • Your environment meets the system requirements to run the Power BI harvester and the lin-

Step	What?	Description	Prerequisites
			eage harvester. Tip For a full ingestion, we highly recommend to have a Power BI Premium subscription.
7	Download and install the lineage harvester.	You use the lineage harvester to trigger the creation of Power BI assets, their relations and a technical lineage in Data Catalog. You can download the lineage harvester from the Collibra Product Resource Downloads page.	Your environment meets the system requirements to install and use the lineage harvester.

Step	What?	Description	Prerequisites
8	Prepare the lineage harvester configuration file and run the lineage harvester.	You create a lineage harvester configuration file with Power BI connection information and run the lineage harvester to import the results of the Power BI integration and the technical lineage for Power BI into Data Catalog. As a result, Collibra creates new Power BI assets in Data Catalog and imports relations between these assets. It also creates a technical lineage for Power BI assets and other data sources in the lineage harvester configuration file. Tip For more information about the lineage harvester, see the Collibra Data Lineage documentation.	• You have downloaded the lineage harvester version 1.2.1 or newer. • Your environment meets the system requirements to install and run the lineage harvester. • You have prepared a Power BI harvester configuration file. • You have a global role with the Catalog global permission, for example Catalog Author. • You have a global role with the Technical lineage global permission. • You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain: ◦ Asset: add ◦ Attribute: add ◦ Domain: add ◦ Attachment: add

Step	What?	Description	Prerequisites
9	View the Power BI assets and technical lineage	After the Power BI metadata is ingested in Data Catalog, you can go to the domain where you ingested Power BI and see the list of ingested Power BI assets. These assets are automatically stitched to existing assets in Data Catalog. You can go to a Power BI Column asset page and click the Technical lineage lineage tab to view the technical lineage. Note If you ingest Power BI for the first time or if you change your geolocation or cloud provider, you have to restart the DGC service before you can see your technical lineage. Warning When you run the harvesters, Collibra Data Lineage creates all Power BI assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize	You have a Data Catalog global role with the Technical lineage global permissions.

Step	What?	Description	Prerequisites
		Power BI. As a consequence, all manually added data of those assets is lost.	

Note The order in which you run the harvesters is important. You first have to run the Power BI harvester to collect the metadata from your Power BI application and then run the lineage harvester to import new Power BI assets and their relations in Data Catalog.

Ingestion results based on Power BI subscriptions

You can only ingest Power BI metadata to which the Power BI user has access to. Your level of access is determined by your Power BI subscriptions.

The following table gives an overview of the minimum required subscriptions and the results in Data Catalog.

Note

- The assets have the same names as their counterparts in Power BI. Full names and Display names cannot be changed in Data Catalog.
- Depending on your Power BI subscription, it could be that not all asset types are created.
- Asset types are only created if you have all specific Power BI and Data Catalog permissions.
- All Power BI asset types are created in the same domain.
- Relations that were manually created between Power BI assets and other assets in accordance with the relation types in the Power BI operating model, are deleted after a refresh of the Power BI metadata.

Minimum required subscription	Result in Data Catalog
Power BI Pro	Power BI Workspaces are not assigned to a dedicated capacity in Power BI. The following asset types are created in Data Catalog: • Power BI Server • Power BI Workspace • Power BI Dashboard • Power BI Tile • Power BI Report • Power BI Data Model Technical lineage is unavailable.
Power BI Pro with Power BI Embedded Capacity subscription in Microsoft Azure	Power BI Workspaces are assigned to a embedded capacity in Azure. The following asset types are created in Data Catalog: • Power BI Server • Power BI Capacity • Power BI Workspace • Power BI Dashboard • Power BI Tile • Power BI Report • Power BI Data Model • Power BI Table • Power BI Column A technical lineage is created for all Power BI Column assets.

Minimum required subscription	Result in Data Catalog
Premium	Power BI Workspaces are assigned to a dedicated capacity in Power BI. The following asset types are created in Data Catalog: • Power BI Server • Power BI Capacity • Power BI Workspace • Power BI Dashboard • Power BI Tile • Power BI Report • Power BI Data Model • Power BI Table • Power BI Column A technical lineage is created for all Power BI Column assets.

Minimum required subscription	Result in Data Catalog
Premium Per User	Power BI Workspaces are assigned to a dedicated capacity in Power BI. The following asset types are created in Data Catalog: • Power BI Server • Power BI Capacity • Power BI Workspace • Power BI Dashboard • Power BI Tile • Power BI Report • Power BI Data Model • Power BI Table • Power BI Column A technical lineage is created for all Power BI Column assets. > Note The Power BI Premium Per User license is a new license type that is released for general availability, but is still in its preview period. For more information, see the Microsoft documentation.

> Note We highly recommend you to have a Power BI Premium subscription. Power BI Premium also provides additional features and a better speed and performance.

> Warning When you run the harvesters, Collibra Data Lineage creates all Power BI assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Power BI. As a consequence, all manually added data of those assets is lost.

Power BI ingestion limitations

There are a few considerations and limitations that you must take into account when you use the Power BI metadata connector and lineage feature.

Supported subscriptions

> Important Your subscription determines which Power BI metadata the Power BI harvester can collect.

You need one of the following subscriptions to ingest Power BI metadata in Data Catalog:

- Power BI Pro. To ensure a full ingestion, you also need a Power BI Embedded Capacity subscription in Microsoft Azure.
- Power BI Premium.
- Power BI Premium Per User.

  > Note The Power BI Premium Per User license is a new license type that is released for general availability, but is still in its preview period. For more information, see the Microsoft documentation.

Other Power BI subscriptions are currently not supported.

Power BI metadata

The Power BI harvester can only partially access metadata of the following Power BI elements:

- Classic Power BI workspaces, which include My Workspace. Only a full ingestion of new Power BI workspaces is supported.
- Power BI workspaces that are not part of a dedicated capacity.
- Descriptions of most Power BI elements.
- Power BI apps.They can be ingested as Power BI Reports, but there is no easy way to distinguish them from real Power BI reports.

The Power BI harvester cannot access metadata of the following Power BI elements:

- Dataflows.
- Tile subtitles.
- Data from external sources supplying the input for the Power Query expressions in Power BI.

> Important The Collibra Data Lineage server can process most, but not all, complex Power BI metadata. This means that the success rate of a Power BI ingestion can be very high, but almost never 100%.

Known issues

The following table presents the known issues of the Power BI integration in Collibra Data Intelligence Cloud.

Known issue	Description
The Power BI harvester shows an `Internal Server Error` because of the Power BI workspace filter.	If you want to ingest a lot of Power BI data and you use the WorkspaceFilter in the Power BI configuration file, the Power BI API can go in timeout and, as a result, you get an `Internal Server Error`. If you get this error, we highly advise to not use the workspace filter. > Note If the error message indicates that the issue is an internal server error, the problem is caused by the Power BI REST API, not the Power BI harvester itself.
The data set *Report Usage Metrics Model* cannot be ingested.	The *Report Usage Metrics Model* is a data set that is automatically created by Power BI. This data set does not contain actual data, which means that they contain nothing to ingest into Data Catalog. However, the Power BI harvester still tries to access the metadata and, since there is nothing to access, shows an error message. All error messages about the *Report Usage Metrics* can be ignored.

Known issue	Description

The `IN` operator is not supported.	Currently, the `IN` operator is not supported. As a result, you cannot use `IN` to filter the `workspaceFilter` property on specific Power BI workspace names in the Power BI harvester configuration file.
	
	For example, you want to filter on two Power BI workspace names. In the configuration file, you can enter the following value of the `workspaceFilter` property to ingest only workspaces with the name "workspace1" or "workspace2
	
	`"workspaceFilter": "name eq 'workspace1' or name eq 'workspace2'"`
	
	However, you cannot ingest Power BI workspaces that have "workspace1" or "workspace2" in their name, because the IN operator is currently **not supported**:
	
	`"name in ('workspace1', 'workspace2')"`
	
	Tip For more syntax examples that you can use in the `workspaceFilter` property, see the README file attached to the Power BI harvester or see the Microsoft documentation.

Stitching results are gray.	Usually, data objects that Collibra Data Lineage stitches to assets in Data Catalog have a yellow background in the technical lineage graph. However, assets of BI sources, for example Power BI, that are stitched to other assets in Data Catalog currently have a gray background. This does not indicate that stitching failed. You can see which assets are stitched on the Stitching tab page.

Known issue	Description
Power BI assets that are moved to a different domain are deleted after synchronization.	When you run the harvesters, Collibra Data Lineage creates all Power BI assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Power BI. As a consequence, all manually added data of those assets is lost.
You have successfully ingested Power BI metadata, but calculated columns are not shown in the Technical lineage or in the browse tab pane.	Calculated columns are virtually the same as a non-calculated columns, with one exception: their values are calculated using DAX formulas and values from other columns. Collibra Data Lineage currently does not support internal transformations via DAX language, and any data objects derived via DAX are not shown in the technical lineage or in the browse tab pane. Currently, only M Query/Power Query expressions are supported.

You are experiencing Power Query parsing errors when ingesting Power BI metadata.	The Power Query parser does not currently support parsing for:
	• The following functions:
	  ◦ MicrosoftAzureConsumptionInsights.Tables
	  ◦ Table.ExpandRecordColumn
	  ◦ Dates Query
	• Transact-SQL statements

Supported data sources in Power BI

Power BI is business intelligence software that can integrate with various data sources. When you ingest Power BI metadata, Collibra Data Lineage tries to automatically stitch this metadata to data sources registered in Data Catalog. It also creates a technical lineage that shows where metadata is used and how it transforms.

The following table shows the supported data source types in Power BI that have been tested.

> Warning Although the following data sources have been tested extensively, there still may be some issues caused by unsupported elements within the data source or limitations in the Power BI integration process.

Power BI data source	Technical lineage	Stitching to registered data sources in Data Catalog
Amazon Redshift	Yes	Yes
Azure Databricks	Yes	Yes
Google BigQuery	Yes	Yes
ODBC	Yes	Yes Important You need to use a Power BI <source ID> configuration file to provide the true system names of the ODBC databases in Power BI. For more information, see Providing ODBC database names in Power BI.
Oracle	Yes	Yes
Snowflake	Yes	Yes
SQL Server	Yes	Yes
Sybase	Yes	Yes

> Note We cannot guarantee that other data sources in Power BI can be stitched successfully.

Providing ODBC database names in Power BI

You can create a technical lineage for ODBC data sources in Power BI. However, ODBC database names often can't be determined. When a database name can't be determined, it's given a substitute name, which is the ODBC connection string.

This substitute name can be seen in the technical lineage, but it is merely a placeholder that doesn't carry any meaning if you're trying to identify the database it represents in the technical lineage. A bigger problem is that if you want to stitch the ODBC database to assets in Data Catalog, the substitute name won't match with any ingested databases, so stitching won't work.

To ensure that the true database names appear in the technical lineage, and to ensure successful stitching, you can use a Power BI <source ID> configuration file to provide the true system names of the ODBC databases in Power BI.

> Tip The name "<source ID>" refers to the value of the `sourceId` property in the Power BI configuration file. If, for example, the value of the `sourceId` property in the Power BI configuration file is `power-bi-source-1`, then the name of your <source ID> configuration file should be *power-bi-source-1.conf*.

Example of the <source ID> configuration file

For each ODBC database in Power BI, add the following content to the JSON file:

```
{
   "found_dbname=DSN_MYDATABASE;found_hostname=ODBC": {
       "dbname": "DB001",
       "schema": "MYSCHEMA",
       "dialect": "oracle",
       "collibraSystemName": "oracle-system-name"
   }
}
```

Property	Description
found_ dbname=<substitute database name>;found_ hostname=<server name>	`found_dbname` is the substitute database name. You need to convert it to uppercase and replace every non-alphanumeric character by an underscore (_). In this example, the substitute name is “dsn=MYDATABASE”, so you should use "DSN_MYDATABASE". Note The substitute name is the ODBC connection string, which can be lengthy when it includes the driver and parameters in full. `found_hostname` should be “ODBC”, but you can also use an asterisk (*).
dbname	The true system name of the ODBC database in Power BI.
schema	The name of the default schema of the ODBC database in Power BI. If no schema is specified and the Power BI harvester fails to find a specific schema, it uses the default schema.
dialect	The dialect of the ODBC connection. The dialect must be one of the supported SQL dialects. If no dialect is specified, “mssql” is used, by default. Tip You can enter one of the following values: • *azure*, for an Azure SQL Server data source. • *bigquery*, for a Google BigQuery data source. • *mssql*, for a Microsoft SQL Server data source. • *oracle*, for an Oracle data source. • *redshift*, for an Amazon Redshift data source. • *snowflake*, for a Snowflake data source. • *sybase*, for a Sybase data source.

Property	Description
collibraSystemName	The system or server name of a database. Important Because you are using a <source> configuration file only for the purpose of providing the true system name of an ODBC database in Power BI, you are not required to: • Set the useCollibraSystemName property in the Power BI configuration file to `true`. • Specify a Collibra system name in the <source ID> configuration file. However, if the useCollibraSystemName property is set to `true` in the Power BI configuration file, then you must specify a Collibra system name in the <source ID> configuration file.

For complete information on working with <source ID> configuration files, see Power BI <source ID> configuration file.

Power BI prerequisites

Before you start the Power BI integration process, you have to perform a number of tasks in Power BI and Microsoft Azure. These tasks, which are performed outside of Collibra, are needed to enable the Power BI harvester to reach your Power BI application and collect its metadata.

The tasks include the following:

- The authentication process.
- The registration of your Power BI application in Microsoft Azure.
- The Power BI dedicated capacities and roles for Power BI workspaces.

The metadata harvesting process explains in detail which prerequisites you need to enable the Power BI harvester to collect the Power BI metadata.

> Note There are some limitations to the metadata harvesting process. Make sure you understand these limitations before you start the harvesting process.

> Warning Because these tasks are performed outside of Collibra, it is possible that the content changes without us knowing. We strongly recommend that you carefully read the source documentation.

Supported Power BI subscriptions

> Important Your subscription determines which Power BI metadata the Power BI harvester can collect.

You need one of the following subscriptions to ingest Power BI metadata in Data Catalog:

- Power BI Pro. To ensure a full ingestion, you also need a Power BI Embedded Capacity subscription in Microsoft Azure.
- Power BI Premium.
- Power BI Premium Per User.

  > Note The Power BI Premium Per User license is a new license type that is released for general availability, but is still in its preview period. For more information, see the Microsoft documentation.

> Tip We highly recommend you to have a Power BI Premium subscription.

Authentication

You have to be authenticated to access Power BI metadata. Your authentication method determines how you retrieve the metadata.The Power BI harvester supports two types of authentication:

- Username and password
- Service principal authentication

The metadata harvesting process is different for each authentication method. As a result, different configurations in Microsoft Azure and Power BI are required.

> Note We recommend that you use the service principal authentication.

Username and password

The username and password authentication method relies on the username, in the form of an email address, and a password you provide to access the Power BI metadata.

To use the username and password authentication, you need to be an Azure Active Directory user with a Power BI admin role in Power BI and have a Contributor role in the Power BI workspaces that you want to ingest into Data Catalog.

When you become an Azure Active Directory user, a new email address is created. You use this email address to sign in to Power BI.

The email address that is created in Microsoft Azure is the username that you use to sign in to Power BI. You can store the username and password you use to sign in to Power BI in the Power BI configuration file.

In the Power BI Tenant settings in Power BI, you have to enable the **Allow XMLA endpoints and Analyze in Excel with on-premises datasets**. This setting has to be applied to the entire organization (default) or to the specific security group to which your workspaces belong.

> Note Only Azure Administrators can create users and require them to authenticate via username and password. The Azure Administrator also assigns the user the Power BI admin role. This user is only created for the purpose of Power BI integration in Collibra Data Intelligence Cloud. The user in Azure should have a Member user type.

Service principal

The Service Principal authentication method lets an Azure Active Directory automatically access Power BI.

The Service Principal authentication relies on the Power BI Tenant ID and the Azure Active Directory application ID that you provide in the configuration file. The password you need to access Power BI is the client secret key of the Azure Active Directory application.

To use the Service principal authentication, you need to embed Power BI content with a Service Principal and an application secret. This means that you do the following:

- Create an Azure AD security group.
- Add the security group in the Power BI Tenant settings in Power BI.
- In the Power BI Admin portal, you also do the following :
  - Enable the **Allow service principals to use read-only Power BI admin APIs (preview)** option.
  - Enable the **Allow service principal to use Power BI APIs** option in the Developer settings.
  - Apply the option to specific security groups.
  - Enter the name of the security group to which you want to add the service principal.
  - Enable the **Allow XMLA endpoints and Analyze in Excel with on-premises datasets**. This setting has to be applied to the entire organization (default) or to the specific security group to which your workspaces belong.

  > Note You need Power BI administrator rights to access the Power BI Admin portal.

- Assign the Contributor role to the security group in the Power BI workspaces you want to ingest.

> Tip Do not confuse the **Allow service principals to use read-only Power BI admin APIs (preview)** option with the **Allow service principal to use Power BI APIs** option. You need to enable both options.

Register Power BI in Microsoft Azure and set permissions

Before you set up the Power BI harvester, make sure that the harvester can reach Power BI by registering Power BI in Azure and setting the necessary permission to harvest the

metadata.

We highly recommend that you read about supported authentication methods before you register Power BI in Microsoft Azure.

> Warning This procedure is performed outside of Collibra. A third party may change the software without notification, which can render this documentation out of date. We highly recommend that you carefully read the source documentation.

Steps

> Tip The content in this topic is different for the username / password authentication method or service principal authentication method. We highly recommend that you read the following instructions carefully before you register Power BI in Microsoft Azure:
>
> - Service principal instructions
> - Username / password instructions

1. Register Power BI in the Azure Portal using the following settings:

Setting	Description
Name	The name of your Power BI application.
Supported account types	The type of tenant. This indicates who can access the Power BI application. In this case, the supported account type must be *Single tenant*.
Redirect URI	The location to which a user's client is redirected and where security tokens are sent after a successful authorization. In this case, the redirected URI must be *Web*, but you do not have to specify any web location.

   » When you have registered Power BI, the Azure portal creates two important IDs that you need in the Power BI configuration file:
   - The Application (client) ID
   - The Directory (tenant) ID

   > Note We highly recommend that you store these IDs for further use. You can find the IDs in the **Overview** pane on the Azure portal or in the top right menu.

2. Create a user with the Power BI Administrator role (only for username / password authentication).
3. In the Azure portal, go to **Authentication** pane and do the following:
   a. Go to the **Advanced settings** section.
   b. Set the **Treat application as a public client** to *Yes*.
4. Go to the **API permissions** pane and do the following:
   a. Select **Delegated permissions** as permission type.
   b. Grant the Power BI application in Microsoft Azure the Microsoft Graph `User.Read` permission.
   c. Grant the Power BI application in Microsoft Azure all Power BI Service permissions (only for username / password authentication).

d. Set **Admin consent required** for `Tenant.ReadAll` permission to *Yes* (only for username / password authentication).

» The user now has the following permissions:

- Microsoft Graph
  - `User.Read`
- Power BI Service (only for username / password authentication)
  - `App.Read.All`
  - `Capacity.Read.All`
  - `Dashboard.Read.All`
  - `Dataflow.Read.All`
  - `Group.Read.All`
  - `Report.Read.All`
  - `Tenant.Read.All`, with **Admin consent required** set to `Yes`.
  - `Workspace.Read.All`

5. In the Power BI Admin portal, do the following (only for service principal authentication):
   a. Enable the **Allow service principals to use read-only Power BI admin APIs (preview)** option.
   b. Enable the **Allow service principal to use Power BI APIs** option in the Developer settings.
   c. Apply the option to specific security groups.
   d. Enter the name of the security group to which you want to add the service principal.
   e. Enable the **Allow XMLA endpoints and Analyze in Excel with on-premises datasets**.
   f. Apply the integration setting to the entire organization (default) or to the specific security group to which your workspaces belong.

> Note You need Power BI administrator rights to access the Power BI Admin portal.

6. In the Power BI Admin portal, do the following (only for username / password authentication):
   a. Enable the **Allow XMLA endpoints and Analyze in Excel with on-premises datasets**.

b. Apply the integration setting to the entire organization (default) or to the specific security group to which your workspaces belong.

What's next?

You can add your Power BI workspaces to a dedicated capacity.

Power BI workspaces

Power BI workspaces represent the most used metadata in Power BI. They contain, for example, reports and data sets. If you want a full ingestion, you have to make sure that the Power BI harvester can access all metadata in your Power BI workspaces. Consider the following:

- All Power BI workspaces that you want to ingest into Data Catalog must be a part of the Power BI Premium dedicated capacity.
- Depending on the authentication type, you must have specific roles and permissions to access the metadata in the Power BI workspaces.
- You can only fully ingest new Power BI workspaces. This means that classic workspaces and My Workspace in Power BI are not supported.

> Tip You can filter on Power BI workspaces in the Power BI configuration file.

Power BI Premium dedicated capacity

The Power BI harvester accesses data sets in Power BI's dedicated capacity through XMLA endpoints. As a result, you must add all Power BI workspaces that you want to ingest into Data Catalog to the dedicated capacity.

The following subscriptions offer access to dedicated capacities:

- A Power BI Premium subscription.
- A Power BI Premium Per User subscription.

  > Note The Power BI Premium Per User license is a new license type that is released for general availability, but is still in its preview period. For more information, see the Microsoft documentation.

- A Power BI Pro subscription with Power BI Embedded subscription in Microsoft Azure

  > Tip If you have a Power BI Pro subscription, you don't automatically have access to dedicated capacities. As a result, some metadata may be skipped when harvesting the Power BI metadata and not all Power BI assets are created in Data Catalog. You can prevent that by purchasing a Power BI Embedded subscription in Microsoft Azure or upgrading to Power BI Premium.

Once you have access to dedicated capacities, you can add the Power BI workspaces that you want to ingest to them:

- Via the Power BI Admin portal.
- Via the Power BI workspace in Power BI.

  > Tip For more information about adding Power BI workspaces to a dedicated capacity, see the Power BI documentation.

Roles and permissions

Depending on the authentication type that you want to use, you also require additional permissions in the Power BI workspaces to access the Power BI metadata.

- In case of username / password authentication, the Azure Active Directory user with a Power BI admin role in Power BI must have the Contributor role in the Power BI workspaces you want to ingest.
- In case of Service Principal authentication, you have to add the Active Directory security group to which you added the Service Principal to your Power BI workspaces. The Power BI workspaces you want to ingest must have the Contributor role in the Power BI security group.

Ingesting deleted workspaces

If you delete a Power BI workspace, the workspace is maintained for a 90-day grace period, during which a Power BI administrator can restore the workspace. During the grace period, the workspace has the state Deleted. When you ingest Power BI metadata in Data Catalog, this deleted workspace is ingested.

When the grace period elapses, the state of the workspace becomes Removing, for a short time, while it is being permanently removed. The state then becomes Not found. At this point, as the workspace no longer exists in Power BI, the Power BI Workspace asset in Collibra will also be deleted upon the next synchronization.

Why are deleted workspaces ingested?

Let's image that you ingest a Power BI workspace with the Active state and that over time, you add comments, tags and characteristics to the asset in Collibra. Now let's imagine that the workspace is deleted in Power BI and we do not ingest the deleted workspace. In this case, the Power BI Workspace asset in Collibra is deleted upon the next synchronization. But what if the Power BI administrator decides, during the 90-day grace period, to restore the workspace in Power BI? Upon the next synchronization, a new Power BI Workspace asset is created in Collibra, but all of the comments, tags and characteristics that were part of the deleted asset are lost.

By ingesting deleted Power BI workspaces, we safeguard against losing any of the additional information on the Power BI Workspace asset, in case a Power BI administrator decides to restore a workspace during the grace period.

Viewing workspace states in Collibra

On Power BI Workspace asset pages, you can include the attribute type State, to show the state of ingested Power BI workspaces. To do so, you have to edit the global assignment of the Power BI Workspace asset type and assign the attribute type State.

For complete information on Power BI workspaces and possible states, see the Microsoft Power BI documentation.

The metadata harvesting process

Collibra uses two methods to harvest Power BI metadata: via REST API calls and via XMLA endpoints. The REST API retrieves basic metadata, and XMLA endpoints retrieve more specific metadata.

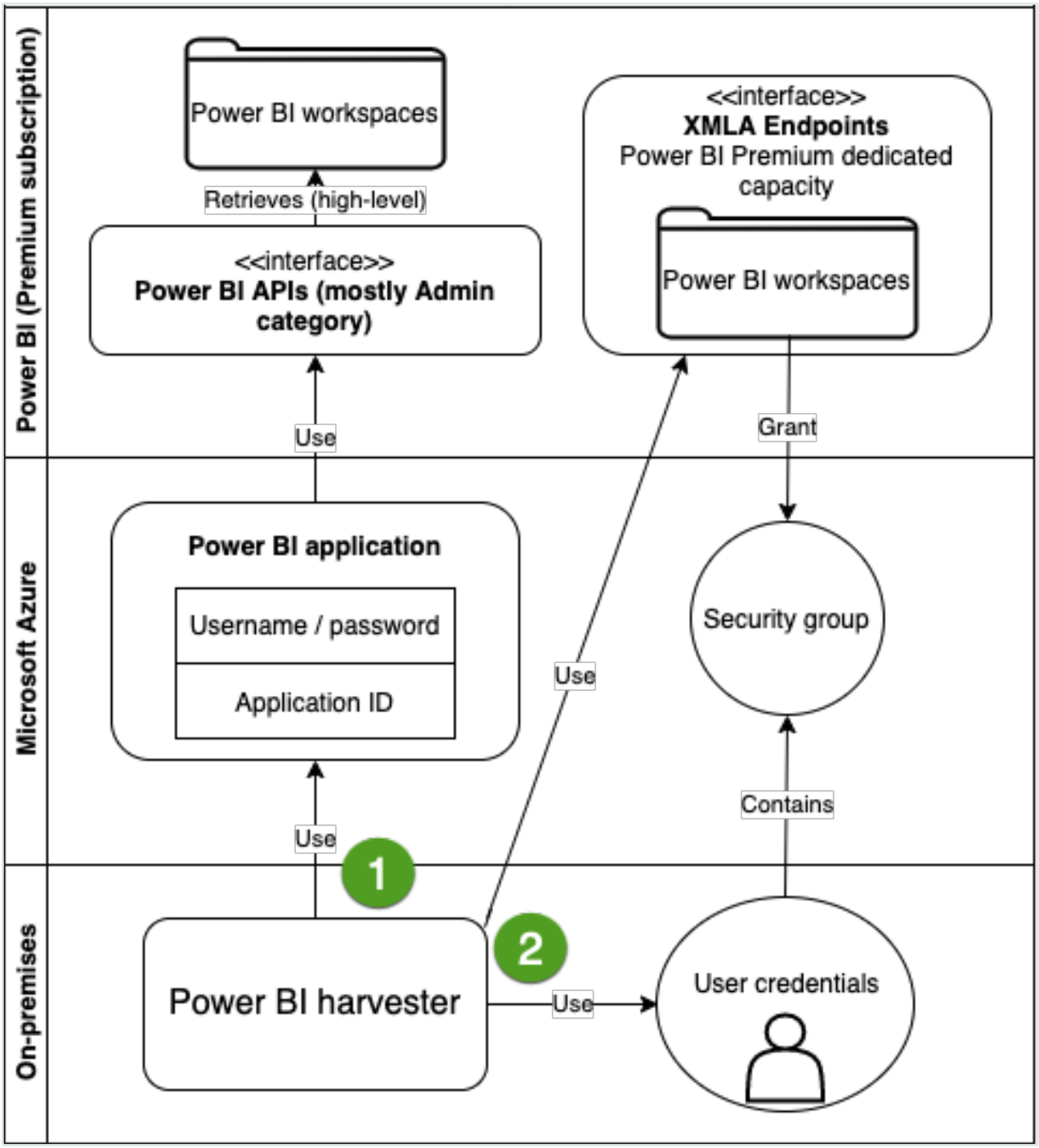
To enable the lineage harvester to access metadata in Power BI workspaces, you must add the workspaces to a Power BI Premium dedicated capacity and have the correct configurations in Microsoft Azure.

> Note There are some limitations to the metadata harvesting process. Make sure you understand these limitations before you start the harvesting process.

The following table shows which metadata the Power BI harvester retrieves and how.

Metadata about...	is retrieved using...
Reports	Microsoft Azure Admin Power BI REST API calls.
Data set columns and lineage	XMLA (Queries or M-queries) endpoints

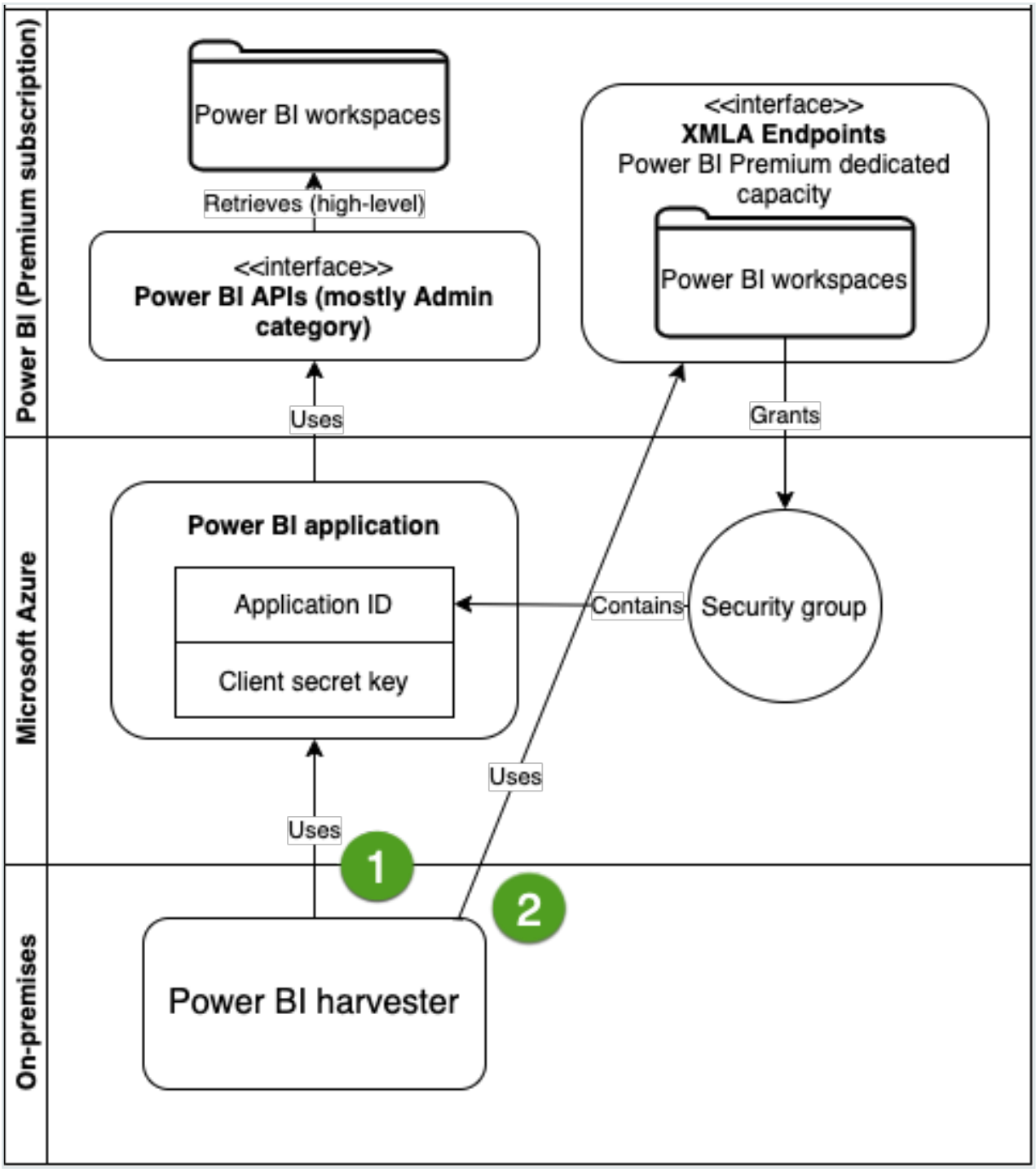
Overview of the metadata harvesting process with username / password authentication



Step	Retrieved via	Description
1	Power BI API calls	The Power BI harvester uses the username, password and application ID to access the Power BI APIs. These APIs retrieve basic Power BI metadata, for example metadata in the Power BI tenant or server and reports.
2	XMLA	You add the Azure Active Directory user with a Power BI admin role in Power BI to a security group and grant him the Contributor role in Power BI workspaces. You add the Power BI workspaces that you want to ingest to the same security group. As a result, the Power BI harvester uses XMLA endpoints to retrieve more specific metadata, for example Power BI columns and lineage. Specific metadata from Power BI workspaces is only harvested if you added the Power BI workspaces to the Power BI dedicated capacity and you have the necessary permissions to harvest the metadata..

> Note Make sure that all necessary dedicated capacities are running and accessible to the Power BI harvester. If not, creating assets for Power BI data sets and your technical lineage may fail.

Overview of the metadata harvesting process with service principal authentication



Step	Retrieved via	Description
1	Power BI API calls	The Power BI harvester uses the application ID and the client secret key of the Azure Active Directory application to access the Power BI APIs. These APIs retrieve basic Power BI metadata, for example metadata in the Power BI tenant or server and reports.
2	XMLA	You add the service Principal to a security group and grant it the Contributor role in the Power BI workspaces. As a result, the Power BI harvester uses XMLA endpoints to retrieve more specific metadata, for example in Power BI columns and lineage. Specific metadata from Power BI workspaces is only harvested if you add the Power BI workspaces to the dedicated capacity and you have the necessary permissions to harvest the metadata.

> Note Make sure that all necessary dedicated capacities are running and accessible to the Power BI harvester. If not, creating assets for Power BI data sets and your technical lineage may fail.

Prepare a domain for Power BI ingestion

You can create a new domain for your Power BI asset and use the domain ID in the Power BI harvester configuration file. As a result, Collibra uses this domain to ingest all Power BI assets during the Power BI integration process.

Prerequisites

- You have a resource role with the Domain > Add resource permission.

Steps

1. In the main menu, click the **Create** (+) button.
   » The **Create** dialog box appears.
2. Click the **Organization** tab.
3. Click a domain type from the list.
   If you clicked the wrong domain type here, you can change it in the **Type** field in the next screen.
   » The **Create Domain** dialog box appears.
4. Enter the required information.

Field	Description
Type	The domain type of the domain you are creating. In this case, you need to select *BI Catalog*.
Community	The community under which the domain will be located.
Name	The name of the new domain.

5. Click **Create**.
6. Open your domain.
7. Copy the domain ID.

> Tip If you go to your domain, you can find the domain ID in the URL. The URL looks like: https://<yourcollibrainstance>/domain/**22258f64-40b6-4b16-9c08-c95f8ec0da26**?view=00000000-0000-0000-0000-000000040001. In this example, the domain ID is in bold.

8. Paste the domain ID in the Power BI configuration file.

> Warning When you run the harvesters, Collibra Data Lineage creates all Power BI assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Power BI. As a consequence, all manually added data of those assets is lost.

Power BI and lineage harvester set-up

To ingest Power BI metadata in Data Catalog, you need to run two different harvesters:

- The Power BI harvester
- The lineage harvester.

The order in which you run the harvesters is important. You first have to run the Power BI harvester to collect the metadata from your Power BI application and then run the lineage harvester to import new Power BI assets and their relations in Data Catalog. The Power BI ingestion workflow explains which roles the harvesters play in the Power BI ingestion process.

> Note You need to purchase the Power BI metadata connector and lineage feature to access the Power BI and lineage harvesters.

> Warning If you upgrade from Power BI harvester 1.0.0.0 to Power BI harvester 1.0.0.1 or newer, you have to follow an upgrade procedure.

Power BI ingestion workflow

To ingest Power BI metadata into Data Catalog, you use two types of harvesters:

- A Power BI harvester
- A lineage harvester

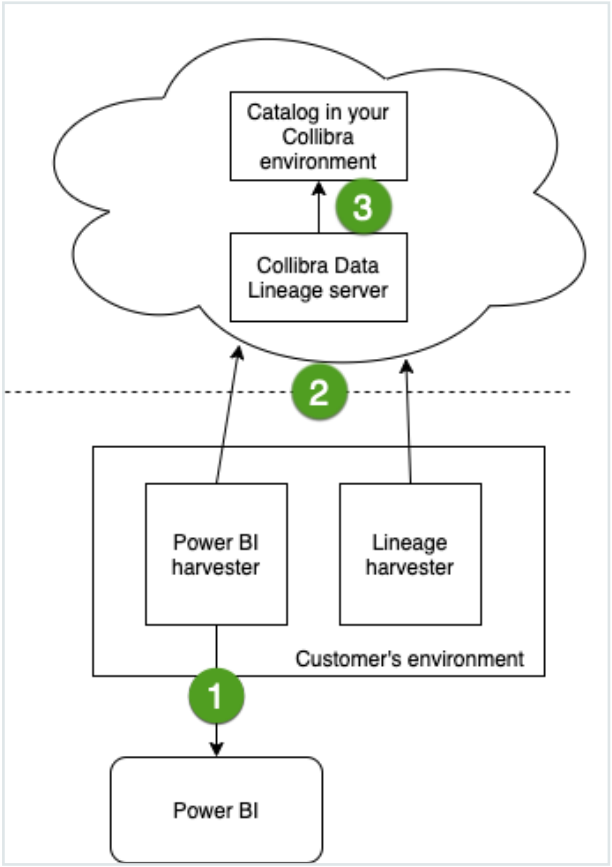
> Note The harvesters can run on the same or on different machines. However, the Power BI harvester must run on a Windows machine.

When the Power BI harvester initiates the Power BI integration, each workflow component performs the following actions:

1. The Power BI harvester:
   - Communicates with Power BI.
   - Harvests Power BI metadata for ingestion and lineage.
   - Sends the Power BI metadata to the Collibra cloud environment.

   > Note The Power BI harvester only harvests the metadata, it does not change it.

2. The lineage harvester:
   - Triggers a new synchronization of the metadata in Collibra to create a technical lineage for Power BI and new relations between Power BI assets.
   - Sends the Power BI ingestion results to Data Catalog.
   - Sends the lineage results to Data Catalog.
3. Data Catalog (via the Collibra Data Lineage server):
   - Shows the new Power BI assets.
   - Shows a Technical lineage tab on Power BI Column pages.



Collibra Data Lineage servers

A Collibra Data Lineage server processes and analyzes the harvested metadata and uploads it to Data Catalog. Collibra Data Lineage servers never process actual data.

Based on your geographical location and cloud provider, the Power BI harvester sends metadata to one of the following Collibra Data Lineage servers:

- 18.198.89.106 (techlin-aws-eu)
- 54.242.194.190 (techlin-aws-us)
- 15.222.200.199 (techlin-aws-ca)
- 35.205.146.124 (techlin-gcp-eu)
- 34.73.33.120 (techlin-gcp-us)
- 35.197.182.41 (techlin-gcp-au)
- 34.152.20.240 (techlin-gcp-ca)
- 51.105.241.132 (techlin-azure-eu)
- 20.102.44.39 (techlin-azure-us)

> Important You have to whitelist all Collibra Data Lineage servers in your geographic location. For example, if your data is located in Europe, you have to whitelist the following Collibra Data Lineage servers: techlin-aws-eu and techlin-gcp-eu. In addition, we highly recommend that you always whitelist the techlin-aws-us Collibra Data Lineage server as a backup, in case the lineage harvester cannot connect to other Collibra Data Lineage servers.

Set up the Power BI harvester

The Power BI harvester is a standalone console application that runs on a Windows machine. You use it to extract data from the Power BI REST API and XMLA endpoints and send it to the Collibra Data Lineage server in Collibra's cloud environment for analysis.

The metadata harvesting process explains in detail which prerequisites you need to enable the Power BI harvester to collect the Power BI metadata.

> Note There are some limitations to the metadata harvesting process. Make sure you understand these limitations before you start the harvesting process.

Power BI harvester system requirements

You need to meet the following system requirements to install and run the Power BI harvester on your Windows machine.

> Note If you want to successfully ingest Power BI metadata into Data Catalog, you need to meet both the system requirements to run the Power BI harvester, and also the system requirements to run the lineage harvester.

Software requirements

You need to meet the software requirements to install and run the Power BI harvester.

Minimum software requirements

You need the following minimum software requirements:

- Microsoft .NET Framework 4.7.2.
- One of the following:
  - Client operating system: Windows 7 SP1, 8.1 or 10, version 1607.
  - Server operating system: Windows Server 2008 R2 SP1.

> Note .NET Framework 4.7.2 is available as a system update.

Recommended software requirements

The minimum software requirements are most likely insufficient for production environments. We recommend you meet the following software requirements:

- Microsoft .NET Framework 4.7.2 or higher.
- Client operating system: Windows 10 April 2018 update, version 1803 or newer.
- Server operating system: Windows Server 2016 version 1803 or newer.

Hardware requirements

You need to meet the hardware requirements to install and run the Power BI harvester.

Minimum hardware requirements

You need the following minimum hardware requirements:

- 2 GB RAM
- 1 GB free disk space

Recommended hardware requirements

The minimum requirements are most likely insufficient for production environments. We recommend you meet the following hardware requirements:

- 4 GB RAM
- 20 GB free disk space

Network requirements

You have firewalls rule to have access to:

- The Microsoft API.
- A Collibra Data Lineage server with IP address:
  - 18.198.89.106 (techlin-aws-eu)
  - 54.242.194.190 (techlin-aws-us)
  - 15.222.200.199 (techlin-aws-ca)
  - 35.205.146.124 (techlin-gcp-eu)
  - 34.73.33.120 (techlin-gcp-us)
  - 35.197.182.41 (techlin-gcp-au)
  - 34.152.20.240 (techlin-gcp-ca)
  - 51.105.241.132 (techlin-azure-eu)
  - 20.102.44.39 (techlin-azure-us)

> Note The Power BI harvester connects to different servers based on your geographic location and cloud provider. If your location or cloud provider changes, the Power BI harvester rescans all your Power BI metadata. You have to whitelist all Collibra Data Lineage servers in your geographic location. In addition, we highly recommend that you always whitelist the techlin-aws-us server as a backup, in case the Power BI harvester cannot connect to other Collibra Data Lineage servers.

> Note The Power BI harvester uses port 443 (tcp only).

Install the Power BI harvester

Before you can use the Power BI harvester, you need to download it and install it on your Windows machine. You can download the Power BI harvester from the Collibra Product Resource Center downloads page.

> Warning If you upgrade to Power BI harvester 1.0.0.1 or newer, you have to follow an upgrade procedure.

Prerequisites

- You have Collibra Data Intelligence Cloud 2020.11 or newer.
- You have access to the Power BI harvester on the Downloads page.
- Your environment meets the system requirements to install and use the Power BI harvester.
- You have added Firewall rules so that the Power BI harvester can connect to the Collibra Data Lineage server with one of the following IP addresses:
  - 18.198.89.106 (techlin-aws-eu)
  - 54.242.194.190 (techlin-aws-us)
  - 15.222.200.199 (techlin-aws-ca)
  - 35.205.146.124 (techlin-gcp-eu)
  - 34.73.33.120 (techlin-gcp-us)
  - 35.197.182.41 (techlin-gcp-au)
  - 34.152.20.240 (techlin-gcp-ca)

- 51.105.241.132 (techlin-azure-eu)
- 20.102.44.39 (techlin-azure-us)

Steps

1. Download the Power BI harvester.
2. Unzip the archive.
3. Open the Power BI harvester folder.
   » The Power BI harvester folder contains two folders, a BAT file that you use to run the harvester and a TXT file with information about the configuration file.
   » An empty Power BI configuration file is available in the **config** folder.

What's next?

You can now prepare the Power BI connection properties in the configuration file and run the Power BI harvester.

> Note We highly recommend you to run the Power BI harvester via command line. This enables you to follow the metadata upload and see possible errors that may occur.

Prepare the Power BI configuration file

You create a configuration file for the Power BI metadata that you want to ingest. This configuration file is used by the Power BI harvester to retrieve metadata from Power BI and send it to Collibra to be scanned, processed and analyzed.

Prerequisites

- You have access to the Power BI harvester on the Downloads page.
- You have completed all prerequisite tasks.
- You have a dedicated domain to ingest the Power BI assets.
- You have a global role with the Catalog global permission, for example Catalog Author.
- You have a global role with the Technical lineage global permission.

- You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain:
  - Asset: add
  - Attribute: add
  - Domain: add
  - Attachment: add
- Your environment meets the system requirements to run the Power BI harvester and the lineage harvester.

> Tip For a full ingestion, we highly recommend to have a Power BI Premium subscription.

Steps

1. In the Power BI harvester folder, open the empty configuration file.
2. Enter the values for each property.

Properties	Description	Mandatory
powerbi	This section contains information that is necessary to connect to your Power BI application.	Yes
tenantDomain	The Power BI tenant domain is the domain associated with the Microsoft Azure tenant. This domain is either a default domain or a custom domain. For example, *collibrapowerbi.onmicrosoft.com*. > Note Usually, you can find a list of Power BI tenant or server domains in your Azure Active Directory or in the top right menu.	Yes

Properties	Description	Mandatory
applicationId	The unique ID of the Microsoft Azure Application (client) ID.	Yes
userName	The username that you use to access Power BI. Depending on your authentication type, the username should have a different value: ◦ For username and password authentication, you enter the username that you use when you sign in to Power BI. ◦ For Service Principal authentication, you leave this field empty. Tip If you cannot store your username in the configuration file for security or other reasons, delete this field and provide the username via command line or when prompted by the Power BI harvester.	No

Properties	Description	Mandatory
password	The password or client secret key that you use to access Power BI. Depending on your authentication type, the password needs a different value: ◦ For username and password authentication, you enter the password that you use when you sign in to Power BI. ◦ For Service Principal authentication, you enter the Power BI application client secret key. In case the password is an empty string, leave this field empty. Tip If you cannot store your password in the configuration file for security or other reasons, delete this field and provide the password via command line or when prompted by the Power BI harvester.	No

Properties	Description	Mandatory
workspaceFilter	An option to exclude specific Power BI workspaces from the ingestion process. You can add multiple workspaces. For example "workspace1, workspace2, workspace3". If the workspaceFilter field remains empty or is deleted from the configuration file, all accessible Power BI workspaces are processed and ingested. Tip For more information about the query options to filter Power BI workspaces, see the Microsoft documentation. Be aware that the "IN" operator is currently not supported. Important If you use Power BI harvester older than version 1.1.0.0, the `workspaceFilter` property is named `groupFilter`. This change is backward compatible. However, if you download a new Power BI harvester, we highly recommend to update your configuration file.	No
techlin	This section contains information to identify your Power BI metadata on the Collibra Data Lineage server.	Yes

Properties	Description	Mandatory
sourceId	The unique ID of your Power BI metadata. The lineage harvester uses this ID to locate the Power BI metadata on the Collibra Data Lineage server. Tip This value can be anything as long as it is a unique, human readable ID and the same as the value of the `Id` property in the lineage harvester configuration file. The Power BI and lineage harvesters use the ID to identify a batch of data on the Collibra Data Lineage server.	Yes
catalog	This section contains information that is necessary to connect to Data Catalog.	Yes
domainId	The unique resource ID of the domain in Collibra Data Intelligence Cloud in which you want to ingest the Power BI assets. Tip You can find the domain ID by clicking the domain type. Then look in the URL of your browser to find the ID. The URL looks like https://<yourcollibrainstance>/domain/<domain ID>?<view>.	Yes

Properties	Description	Mandatory
url	The URL of your Collibra Data Intelligence Cloud instance. Note You can only enter the public URL of your Collibra Data Intelligence Cloud environment. Other URLs will not be accepted.	Yes
userName	The username that you use to sign in to Collibra. Tip If you cannot store your username in the configuration file for security or other reasons, delete this field and provide the username via command line or when prompted by the Power BI harvester.	No
password	The password that you use to sign in to Collibra. Tip If you cannot store your passwordin the configuration file for security or other reasons, delete this field and provide the password via command line or when prompted by the Power BI harvester.	No

Properties	Description	Mandatory
useCollibraSystemName	Indication whether you want to use the system or server name of a data source to match to the System asset in Data Catalog during automatic stitching. This is useful when you have multiple databases with the same name. By default, the useCollibraSystemName property is set to `False`. If you want to use it, set it to `True`. ◦ If you set the useCollibraSystemName property to `true`, the Power BI harvester reads the <source-ID> configuration file and takes the value in the collibraSystemName property into account. ◦ If you set the useCollibraSystemName property to `false`, the Power BI harvester ignores the collibraSystemName property in the <source-ID> configuration file. Warning Unless you have multiple databases with the same name, we highly recommend that you keep the default value.	Yes

3. Save the configuration file.
4. Trigger the Power BI harvester to upload the Power BI metadata:
   - Run the following command line if your configuration file is in its default location: `.\powerbi-harvester.bat`
   - Launch the path to the Power BI configuration file if you moved the configuration file to a different location: `.\bin\powerbi-harvester.exe .\config\powerbi-harvester.conf`

> Note We highly recommend you to run the Power BI harvester via command line. This enables you to follow the metadata upload and see possible errors that may occur.

5. If the Power BI harvester prompts for credentials, enter them or use command line options to provide them.

   > Note Credentials provided via command line overwrite the credentials in the configuration file.

   » The Power BI harvester collects the Power BI metadata and sends it to the Collibra Data Lineage server. Collibra scans and analyzes the metadata.

> Tip If you want to ingest multiple Power BI applications, create a new configuration file using a unique ID and repeat these steps. In the lineage harvester configuration file, you can add multiple Power BI sections that each refer to a different ID.

> Note If you are not able to run the Power BI harvester, go to the troubleshooting section to resolve your issues.

Example

This example shows a configuration file with the username / password authentication method.

```
{
 "powerbi": {
  "tenantDomain": "<organization.onmicrosoft.com>",
  "applicationId": "<microsoft-azure-id>",
  "userName": "<your-power-bi-email-address>",
  "password": "<password-to-access-power-bi>",
  "workspaceFilter": "workspace-name1", "workspace-name2"
 },
 "techlin": {
  "sourceId" : "<unique-power-bi-ID>"
  },
 "catalog": {
  "domainId": "<your-catalog-domain>",
  "url": "<url-to-collibra>",
```

```
  "userName": "<my-collibra-username>",
  "password": "<my-collibra-password>"
 },
 "useCollibraSystemName": false
}
```

This example shows a configuration file with the service principal authentication method.

```
{
 "powerbi": {
  "tenantDomain": "<organization.onmicrosoft.com>",
  "applicationId": "<microsoft-azure-id>",
  "userName": "",
  "password": "<secret-key>",
  "workspaceFilter": "<filter-workspace-name>"
 },
 "techlin": {
  "sourceId" : "<unique-power-bi-ID>"
  },
 "catalog": {
  "domainId": "<your-catalog-domain>",
  "url": "<url-to-collibra>",
  "userName": "<my-collibra-username>",
  "password": "<my-collibra-password>"
 },
 "useCollibraSystemName": false
}
```

> Warning If you are ingesting a large amount of Power BI data and you use the workspace filter (workspaceFilter), the Power BI harvester might time out, resulting in an Internal Server Error. If you get this error, we highly advise you to not use the workspace filter. See the known issues in Power BI ingestion limitations.

What's next?

You can now download and install the lineage harvester and prepare the lineage harvester configuration file. The lineage harvester triggers Collibra to create new Power BI assets, stitch them and show a technical lineage for them.

To refresh the Power BI metadata in Data Catalog, you can run the Power BI harvester and lineage harvester again or schedule jobs to run them automatically.

Prepare Power BI <source ID> configuration file

The Power BI harvester uses a Power BI configuration file to collect the Power BI data objects. It then sends the metadata to the Collibra Data Lineage server. However, if the `useCollibraSystemName` in the Power BI configuration file is set to `true`, you also have to provide a specific <source ID> configuration file that defines the system name of databases in Power BI.

Collibra Data Lineage uses the system names to match the structure of databases in Power BI to assets in Data Catalog.

> Tip The name "<source ID>" refers to the value of the `sourceId` property in the Power BI configuration file.

Prerequisites

- The `useCollibraSystemName` in the Power BI harvester configuration file is set to `true`.

> Note This is not a prerequisite if you are using a <source ID> configuration file for the purpose of providing the true system names of the ODBC databases in Power BI. In that case, you can set the `useCollibraSystemName` property in the Power BI harvester configuration file to `true`, but it is not mandatory.

Steps

1. Create a new JSON file in the Power BI harvester **config** folder.
2. Give the JSON file the same name as the value of the `sourceId` property in the Power BI configuration file.

   > Example The value of the `sourceId` property in the Power BI configuration file is `power-bi-source-1`. Therefore, the name of your JSON file should be *power-bi-source-1.conf*.

3. For each database in Power BI, add the following content to the JSON file:

Property	Description	Mandatory?
found_dbname=<database name>;found_hostname=<server name>	The database information of supported data sources in Power BI that is typically collected by the Power BI harvester. It describes on which server the database is running (`found_hostname`) and what the name of the database is (`found_dbname`). Tip You can use wildcards to capture multiple connection string combinations: **Show me the supported wildcards** Pattern: * — Description: Matches everything. Pattern: ? — Description: Matches any single character. Pattern: [seq] — Description: Matches any character in "seq". Pattern: [!seq] — Description: Matches any character not in "seq".	Yes
dbname	The name of the database of a supported data source in Power BI.	No

Property	Description	Mandatory?
schema	The name of the default schema of a supported data source in Power BI. If the Power BI harvester fails to find a specific schema, it uses the default schema.	No
dialect	The dialect of the supported data source in Power BI. Tip You can enter one of the following values: ◦ *azure*, for an Azure SQL Server data source. ◦ *bigquery*, for a Google BigQuery data source. ◦ *mssql*, for a Microsoft SQL Server data source. ◦ *oracle*, for an Oracle data source. ◦ *redshift*, for an Amazon Redshift data source. ◦ *snowflake*, for a Snowflake data source. ◦ *sybase*, for a Sybase data source.	No

Property	Description	Mandatory?
collibraSystemName	The system or server name of a database. **Warning** The value of this property must exactly match the name of your System asset in Collibra. **Important** If you are using a <source> configuration file for the purpose of providing the true system name of an ODBC database in Power BI, you are not required to: ◦ Set the useCollibraSystemName property in the Power BI configuration file to `true`. ◦ Specify a Collibra system name in the <source ID> configuration file. However, if the useCollibraSystemName property is set to `true` in the Power BI configuration file, then you must specify a Collibra system name in the <source ID> configuration file.	Yes (unless you are using a <source ID> file to provide the true system names of ODBC databases in Power BI.)

4. Save the <source ID> configuration file.

Example of the <source ID>.conf file

```
{
   "found_dbname=databasename1;found_hostname=*": {
       "dbname": "mssql-database-name",
       "schema": "mssql-schema-name",
```

```
        "dialect": "mssql",
        "collibraSystemName": "mssql-system-name"
    },
    "found_dbname=databasename2;found_hostname=server-name.on-
microsoft.com": {
        "dbname": "oracle-database-name",
        "schema": "oracle-schema-name",
        "dialect": "oracle",
        "collibraSystemName": "oracle-system-name"
    }
}
```

Ingest multiple Power BI applications

You can ingest more than one Power BI application in Collibra. For each Power BI application, you create a separate Power BI configuration file, and then add a section in the lineage harvester configuration file.

Prerequisites

- You have access to the Power BI harvester on the Downloads page.
- You have completed all prerequisite tasks.
- You have a dedicated domain to ingest the Power BI assets.
- You have a global role with the Catalog global permission, for example Catalog Author.
- You have a global role with the Technical lineage global permission.
- You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain:
  - Asset: add
  - Attribute: add
  - Domain: add
  - Attachment: add
- Your environment meets the system requirements to run the Power BI harvester and the lineage harvester.

> Tip For a full ingestion, we highly recommend to have a Power BI Premium subscription.

Steps

1. Prepare the Power BI configuration file for one Power BI application.
2. Run the Power BI harvester.
3. For each additional Power BI application, do the following:
   a. Prepare a new configuration file with the information of the next Power BI application.
      i. Optionally, create a new domain in Data Catalog to ingest the assets of this Power BI application.
      ii. Enter a new source ID that is different from the source IDs of existing Power BI configuration files.
   b. Run the Power BI harvester again.

   > Note Make sure that you refer to the path of this configuration file when you run the Power BI harvester.

   » The Power BI harvester collects the Power BI metadata of each Power BI application and sends it to the Collibra Data Lineage server.
   » Collibra scans and analyzes the metadata.
4. In the lineage harvester configuration file, create a Power BI section for each Power BI application. Use the source ID of each Power BI configuration file as the ID of the Power BI section in the lineage harvester configuration file.
5. Run the lineage harvester to ingest the Power BI metadata in Collibra.
   » The Power BI metadata is ingested in the domain that you specified in the Power BI configuration file.

Example

You have two Power BI applications that you want to ingest. The first Power BI configuration file has source ID `power-bi-app-a`, the second Power BI configuration file has source ID `power-bi-app-b`. The lineage harvester configuration file contains two Power BI sections that each refer to a different source ID.

```
{
   "general": {
       "catalog" : {
           "url" : "https://companydomain.collibra.com",
           "username" : "my-Collibra-username"}
```

```
    },
    "sources" : [
     {
         "type" : "ExistingLineage",
         "id" : "power-bi-app-a"
    }
    {
         "type" : "ExistingLineage",
         "id" : "power-bi-app-b"
    }]
}
```

What's next?

To refresh the Power BI metadata in Data Catalog, you can run the Power BI harvester and lineage harvester again or schedule jobs to run them automatically. You can schedule to synchronize Power BI applications at different times.

Command options and arguments

After creating a Power BI harvester configuration file, you can use the command line to provide the Power BI harvester with additional information or perform specific actions.

> Note Credentials provided via command line overwrite the credentials in the configuration file.

Typical command options and arguments

The following table shows the most commonly used command options and arguments.

Command	Description
`--powerbi-password "<Power BI user password or application client secret key>"`	Your Power BI password. If you don't want to add your password in the Power BI harvester configuration file, you can provide it via command line. Your password depends on the authentication method that you use: • For username / password authentication, you enter the Power BI user password. • For Service Principal authentication, you enter the application client secret.
`--powerbi-user"<Power BI username or empty string>"`	Your Power BI username. If you don't want to add your username in the Power BI harvester configuration file, you can provide it via command line. Your username depends on the authentication method that you use: • For username / password authentication, you enter the Power BI username. • For Service Principal authentication, you enter "" to indicate an empty string. This is only necessary if you deleted the username filed in the configuration file.

Command	Description
`--catalog-password "<Collibra password>"`	Your Collibra password. If you don't want to add your password in the Power BI harvester configuration file, you can provide it via command line. > Note If you added an API key, the Data Catalog credentials will not be used.
`--catalog-user"<Collibra username>"`	Your Collibra username. If you don't want to add your password in the Power BI harvesterconfiguration file, you can provide it via command line. > Note If you added an API key, the Data Catalog credentials will not be used.
`--output-file <file path>`	Save your harvested Power BI metadata to a specified file.
`--from-file <file path>`	Upload Power BI metadata that was already harvested and saved to a specified file.
`--timeout <seconds>`	Increase the timeout duration to specify a longer timeout for remote API calls. > Example If you want the Power BI harvester to wait 15 minutes before canceling a remote API connection, you can use `--timeout 900`.

Set up the lineage harvester for Power BI ingestion

The lineage harvester is a software application that is needed to collect your Power BI metadata and send it to the Collibra Data Lineage server, where the metadata is processed and a technical lineage and new Power BI assets and relations are created. Collibra Data Intelligence Cloud then import those assets and relations into Data Catalog.

For more information about the lineage harvester, read the Collibra Data Lineage documentation.

> Note You need the lineage harvester 1.2.1 or newer to ingest Power BI metadata into Data Catalog.

Lineage harvester system requirements

You need to meet the system requirements to be able to install and run the lineage harvester.

Software requirements

You need the following software requirements to install and run the lineage harvester.

Minimum software requirements

- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.

Recommended software requirements

- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.

Hardware requirements

You need to meet the hardware requirements to install and run the lineage harvester.

Minimum hardware requirements

You need the following minimum hardware requirements:

- 2 GB RAM
- 1 GB free disk space

Recommended hardware requirements

The minimum requirements are most likely insufficient for production environments. We recommend the following hardware requirements:

- 4 GB RAM
- 20 GB free disk space

Network requirements

You need the following minimum network requirements:

- Firewall rules so that the lineage harvester can connect to:
  - Collibra Data Intelligence Cloud.
  - All Collibra Data Lineage servers in your geographic location:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)

> Note The Power BI harvester connects to different servers based on your geographic location and cloud provider. If your location or cloud provider changes, the Power BI harvester rescans all your Power BI metadata. You have to whitelist all Collibra Data Lineage servers in your geographic location. In addition, we highly recommend that you always whitelist the techlin-aws-us server as a backup, in case the Power BI harvester cannot connect to other Collibra Data Lineage servers.

Install the lineage harvester for Power BI ingestion

Before you can use the lineage harvester, you need to download it and install it. You can download the lineage harvester from the Collibra Community downloads page.

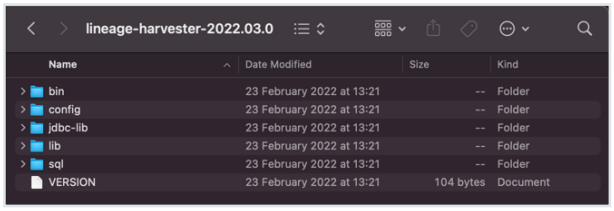
> Warning If you upgrade to lineage harvester 1.3.0 or newer, you have to follow an upgrade procedure.

Prerequisites

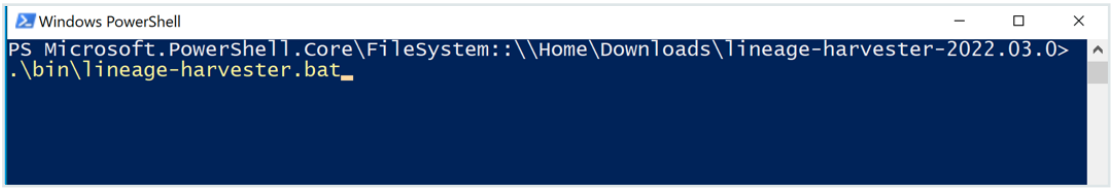
- You have purchased the Power BI metadata connector and lineage feature.
- You have Collibra Data Intelligence Cloud 2020.11 or newer.
- You meet the minimum system requirements.
- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.
- You have added Firewall rules so that the lineage harvester can connect to:
  - the Collibra Data Lineage server with the following IP addresses:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)
  - Collibra Data Intelligence Cloud 2020.11 or newer.

Steps

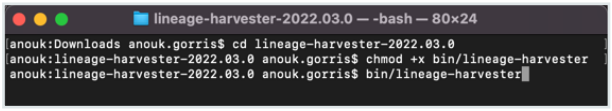
1. Download the lineage harvester version 1.2.1 or newer.
2. Unzip the archive.
   » You can now access the lineage harvester folder.



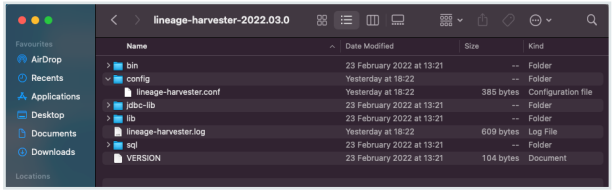
3. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



   » The lineage harvester is installed automatically. You can check the installation by running `./bin/lineage-harvester --help`.

What's next?

You can now prepare the lineage harvester configuration file.

Prepare the lineage harvester configuration file for Power BI

You have to prepare a technical lineage configuration file and run the lineage harvester to fetch the Power BI analysis results on the Collibra Data Lineage server and sent them as an import job to your Collibra Data Intelligence Cloud.

> Note Comments in the lineage harvester configuration file are not supported.

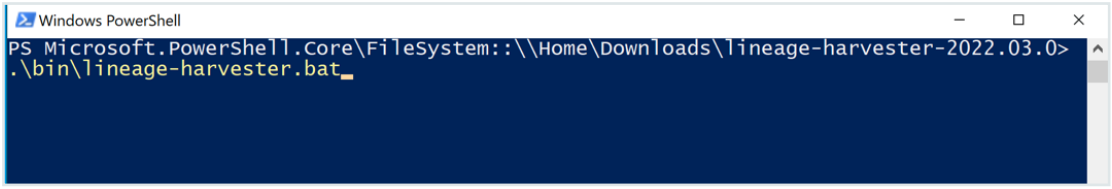
> Tip For more information, see Collibra Data Lineage.

Prerequisites

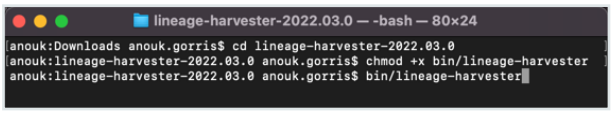
- You have prepared the Power BI configuration file and executed the Power BI harvester.
- You have a global role that has the Manage all resources global permission.
- You have a global role with the Catalog global permission, for example Catalog Author.
- You have the Technical lineage global permission.
- You have created a BI Catalog domain in which you want to ingest the Power BI assets.
- You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain:
  - Asset: add
  - Attribute: add
  - Domain: add
  - Attachment: add
- You have installed the lineage harvester version 1.2.1 or newer and you have the necessary system requirements to run it.

Steps

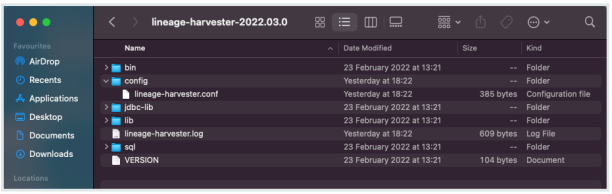
1. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



2. Open the configuration file and enter the values for each property.

Properties	Description
general	This section describes the connection information between the lineage harvester and Data Catalog.
catalog	This section contains information that is necessary to connect to Data Catalog.
url	The URL of your Collibra Data Intelligence Cloud environment. Note You can only enter the public URL of your Collibra DGC environment. Other URLs will not be accepted.

Properties	Description
username	The username that you use to sign in to Collibra.
sources	This section describes the data sources for which you want to create the technical lineage. You have to create a configuration section for each data source. Note You can add multiple data sources to the same configuration file.
type	The kind of data source. In this case, the value has to be *ExistingLineage*.
id	The unique ID to identify the Power BI service metadata that was uploaded to the Collibra Data Lineage server. The value has to be the same as the value you used in the `sourceId` property in the Power BI configuration file. Tip This value can be anything as long as it is a unique ID and the same as the value of the `sourceId` property in the Power BI configuration file. The Power BI and lineage harvesters use the ID to identify a batch of data on the Collibra Data Lineage server.

Tip If you want to ingest multiple Power BI applications, create a separate Power BI configuration file for each Power BI application each with a unique source ID. Duplicate the Power BI section in the lineage harvester configuration file and enter the source ID in the ID property.

3. Save the configuration file.
4. Start the lineage harvester again in the console and run the following command:
   - for Windows: `.\bin\lineage-harvester.bat full-sync`
   - for other operating systems: `./bin/lineage-harvester full-sync`

5. When prompted, enter the passwords to connect to your Collibra Data Intelligence Cloud environment.
   » The password is encrypted and stored in **/config/pwd.conf**

What's next?

The lineage harvester triggers Collibra to import Power BI assets and their relations and create a technical lineage for Power BI Column assets. Collibra also stitches the new Power BI assets to existing assets in Data Catalog.

To refresh the Power BI metadata in Data Catalog, you can run the Power BI harvester and lineage harvester again or schedule jobs to run them automatically.

> Tip You can check the progress of the Power BI ingestion and technical lineage creation in Activities. The **Results** field indicates how many relations were imported into Data Catalog.

> Warning When you run the harvesters, Collibra Data Lineage creates all Power BI assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Power BI. As a consequence, all manually added data of those assets is lost.

Power BI business logic

Power BI business users work with Power BI dashboards and reports to make business decisions. Collibra's Power BI connector and lineage feature offers business users several advantages:

- Easily find certified Power BI content.
- Shop for Power BI reports.
- Trace Power BI metadata to metadata of other data sources.
- Find where content is stored in Power BI.
- Get information about a Power BI Report in a single location.

Power BI asset pages

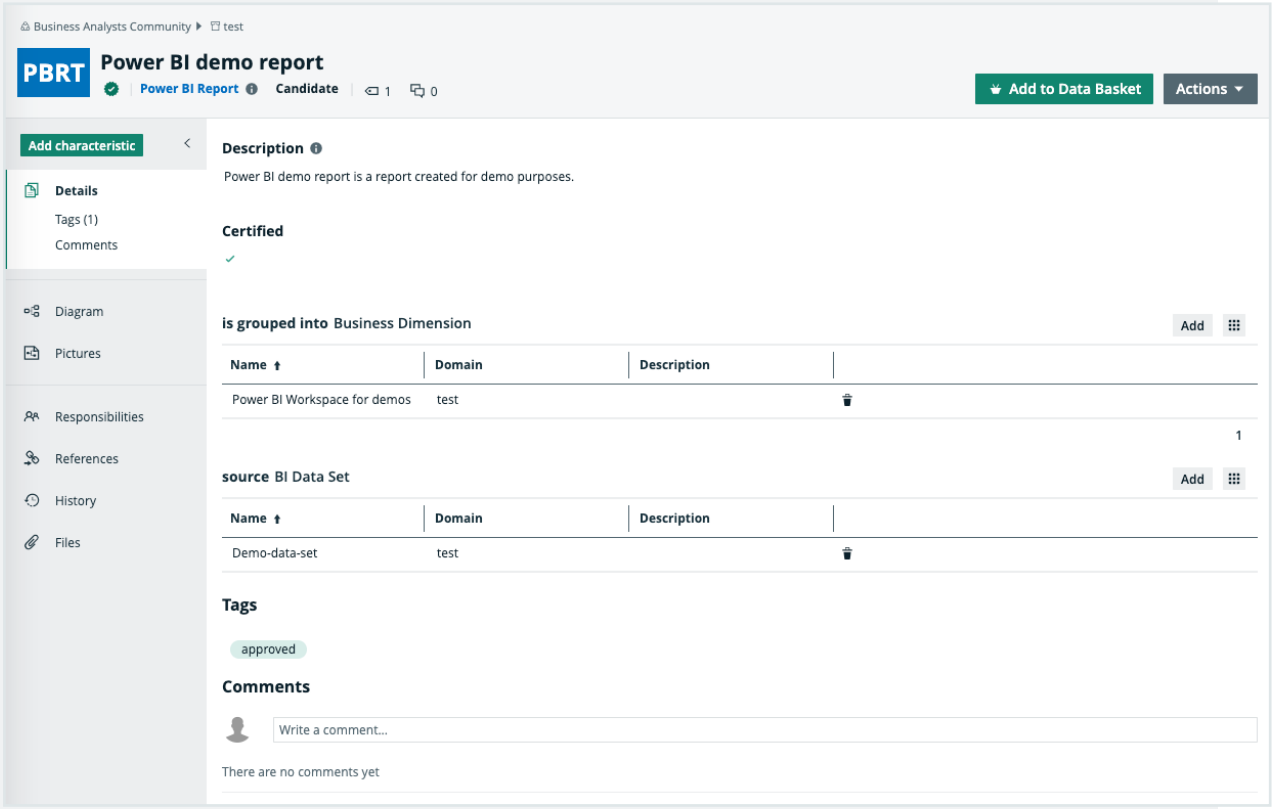
Depending on the Power BI asset type, the asset page shows different information ingested from Power BI. You can find a specific Power BI asset page using Data Catalog search or via the Data Catalog BI domain in which you ingested the Power BI metadata.

Details

Asset pages show attributes and relations to other assets. This information is synchronized with the Power BI service. However, you can add additional characteristics, tags or comments.

If you want to use a Power BI Data Model or a Power BI Report, you can add it to the Data Basket and check it out.

Example The following Power BI Report asset page shows in which Power BI Workspace the report is stored and which Power BI Data Set it uses. This asset has a clear description and is certified.

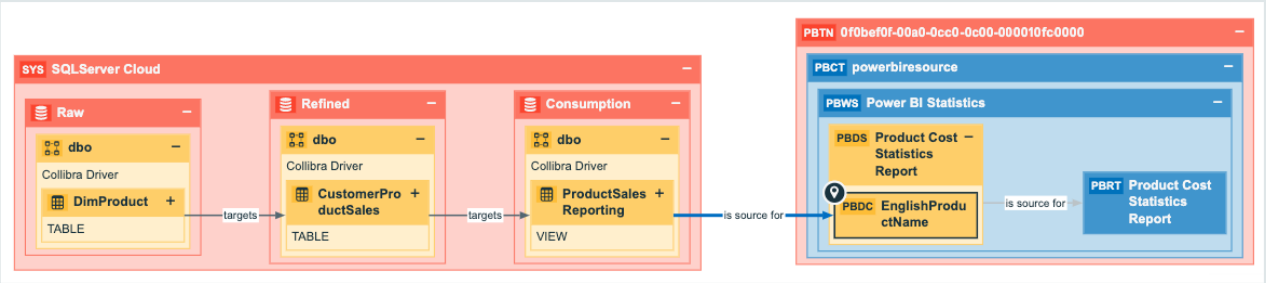


Business diagrams

The business diagram is a feature to show and interact with many assets and relations in an easy-to-read diagram. The business diagram helps you to quickly see to which other assets a specific asset is related. As such, the diagram can show a high-level presentation of a Power BI Report. This enables you, for example, to see:

- In which Power BI Workspace the Power BI Report is stored.
- In which Power BI Capacity the Power BI Workspace is stored.
- Which Power BI Data Model assets the Power BI Report uses.
- Which Table assets and Column assets from other data sources are the source of a Power BI Column asset.

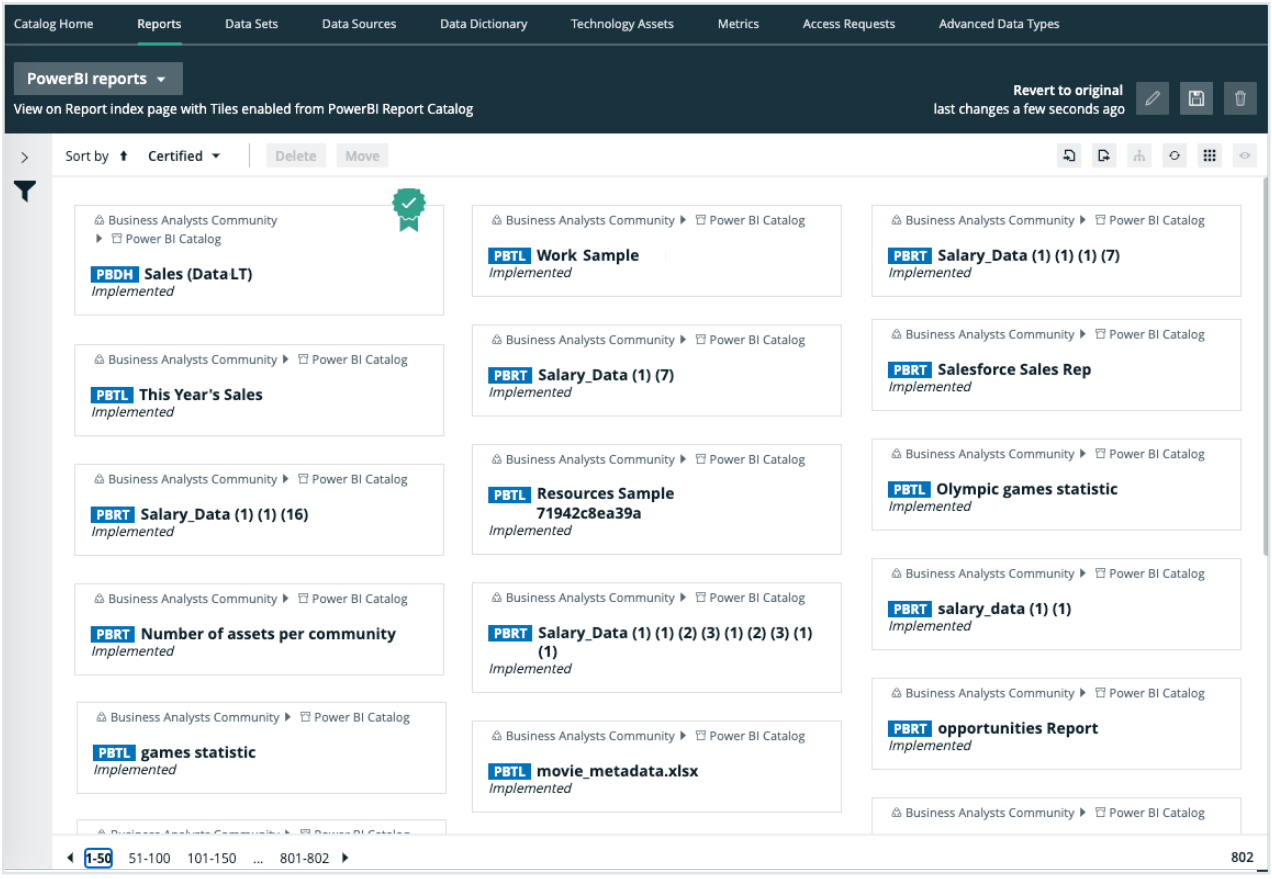
Example The following business diagram shows the *Product Cost Statistics Report* Power BI Report, which is stored in the *Power BI Statistics* Power BI Workspace. The report uses the *Product Cost - Statistics Report* Power BI Data Model. This data set contains data from the *SQL Server Cloud* source.



Report views

The Power BI connector and lineage feature enables you to find all ingested Power BI Reports and children of the Power BI Report asset type in a single location.

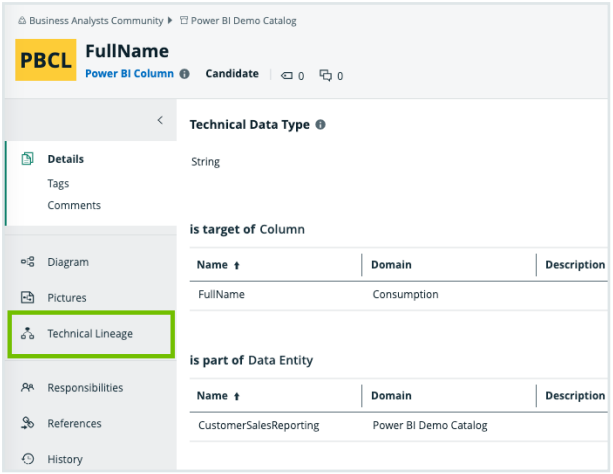
In the **Reports** tab page in Data Catalog you can see an overview of all Report assets and their children. Optionally, you can create a view with a filter to only show Power BI Reports. This is useful if you quickly want to find a report or if you want to know which reports are certified.



Technical lineage for Power BI service

When you ingest Power BI metadata in Data Catalog, you automatically create a technical lineage for Power BI Column assets. Each Power BI Column asset page has a Technical lineage tab page that shows the technical lineage of that Power BI Column asset.

> Note If you ingest Power BI for the first time or if you change your geolocation or cloud provider, you have to restart the DGC service before you can see your technical lineage.



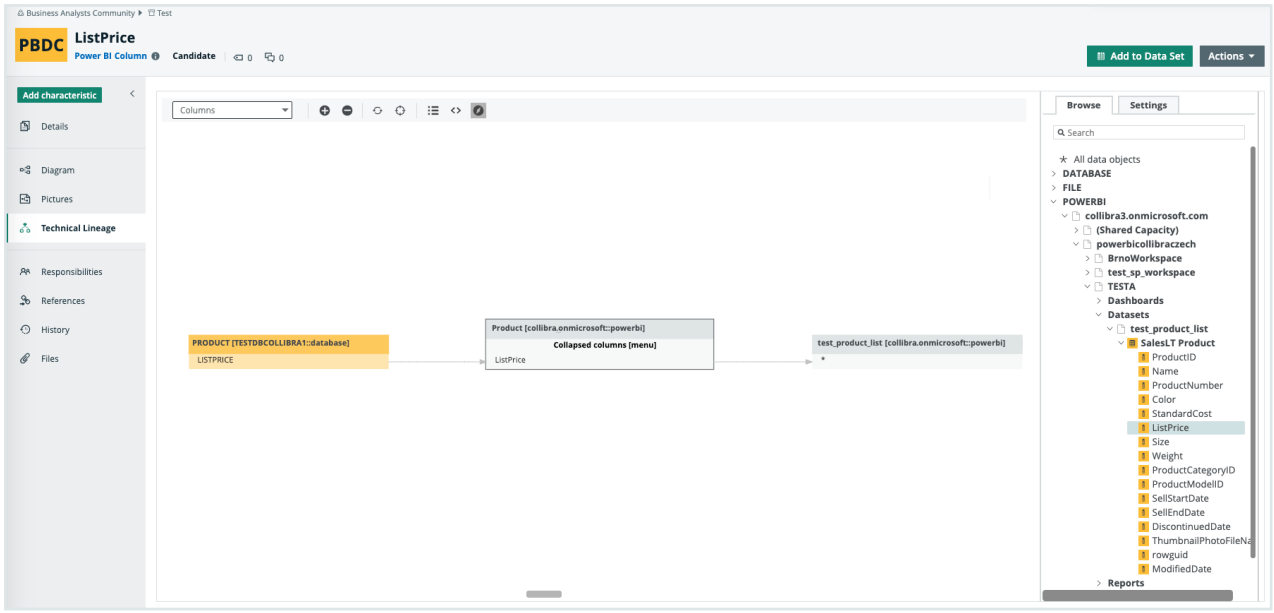
Technical lineage graph

The technical lineage graph shows relations of the type "Data Element targets / sources Data Element" between BI assets and other data objects in the data flow, for example Column assets or Power BI Column assets. These relations are created during the Power BI ingestion process as a result of automatic stitching.

For more information about the technical lineage, see the Collibra Data Lineage section in the user guide.

Example

The following technical lineage shows the relation of the type "Data Element targets / sources Data Element" between the Column asset *LISTPRICE* and the Power BI Column asset *ListPrice*.



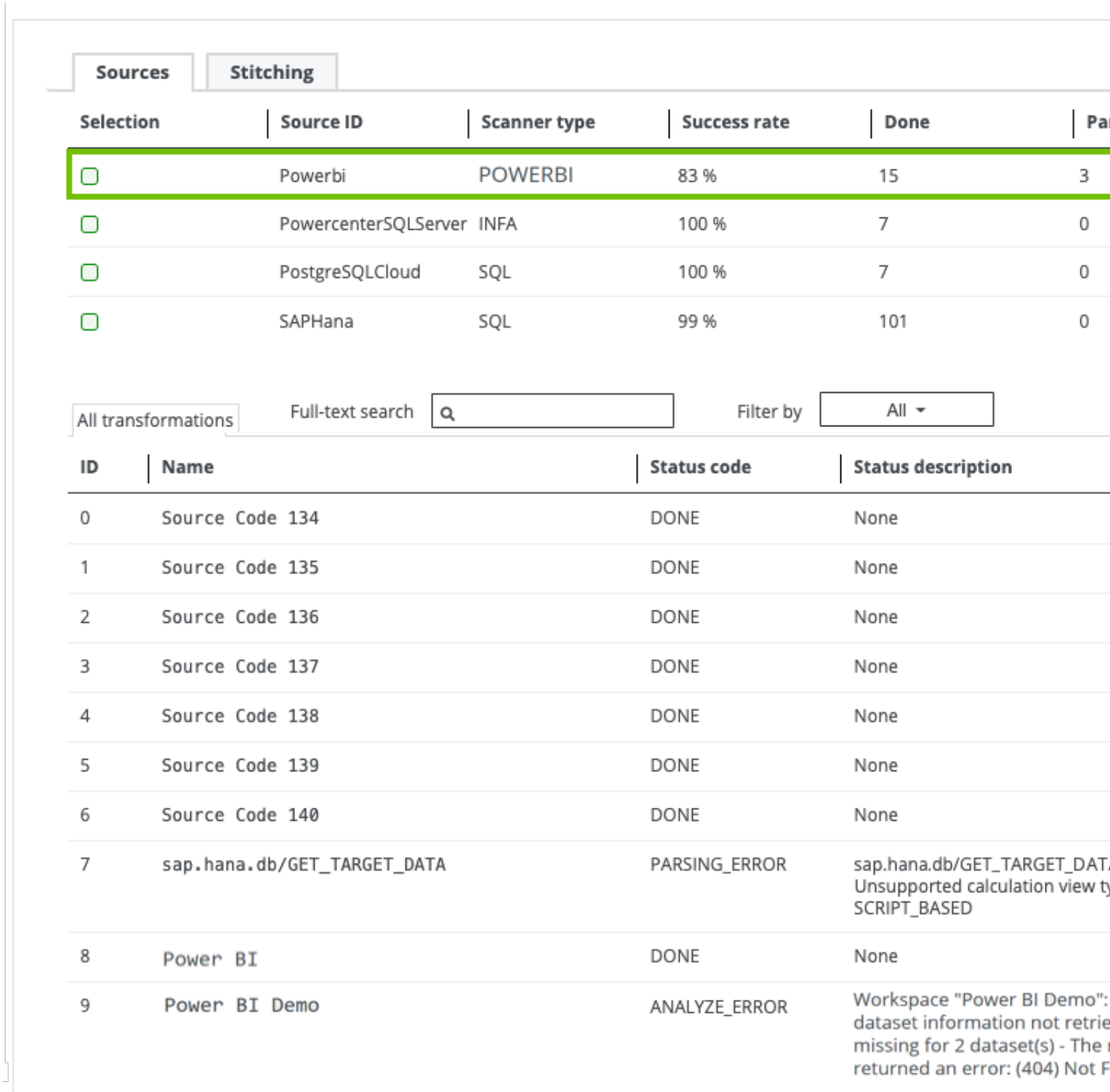
Sources tab page

The Sources tab page shows the transformation details that were analyzed and processed on the Collibra Data Lineage server and the results of this analysis. The success rate of the analysis indicates how complete the technical lineage is. There are a few limitations that prevent the Collibra Data Lineage server from processing all Power BI metadata.

> Important The Collibra Data Lineage server can process most, but not all, complex Power BI metadata. This means that the success rate of a Power BI ingestion can be very high, but almost never 100%.

Example

The following Sources tab page shows that you have created a technical lineage for four data sources. Power BI has a success rate of 83%. When you use the transformation logs to investigate the errors, you see that the Collibra Data Lineage server couldn't process some elements of the Power BI metadata, for example because they are not supported or there is an issue in the configuration file or the Power BI setup.



Sources | Stitching

Selection	Source ID	Scanner type	Success rate	Done	Pa
☐	Powerbi	POWERBI	83 %	15	3
☐	PowercenterSQLServer	INFA	100 %	7	0
☐	PostgreSQLCloud	SQL	100 %	7	0
☐	SAPHana	SQL	99 %	101	0

All transformations | Full-text search | Filter by: All

ID	Name	Status code	Status description
0	Source Code 134	DONE	None
1	Source Code 135	DONE	None
2	Source Code 136	DONE	None
3	Source Code 137	DONE	None
4	Source Code 138	DONE	None
5	Source Code 139	DONE	None
6	Source Code 140	DONE	None
7	sap.hana.db/GET_TARGET_DATA	PARSING_ERROR	sap.hana.db/GET_TARGET_DAT Unsupported calculation view t SCRIPT_BASED
8	Power BI	DONE	None
9	Power BI Demo	ANALYZE_ERROR	Workspace "Power BI Demo": dataset information not retrie missing for 2 dataset(s) - The returned an error: (404) Not F

Automatic stitching

Stitching is a process that creates relations between database columns that are Column assets in Collibra Data Intelligence Cloud and BI assets representing the same database,

specifically between:

- The assets that are created when you ingest Power BI.
- The assets that are created when you register a data source.

The Power BI Harvester collects the Power BI source code and sends it to Collibra for analyzing. The lineage harvester then pushes it to the Data Catalog and creates the relation between Power BI assets in Data Catalog.

At the same time, Collibra analyzes other metadata of data sources that you registered in Data Catalog and creates new relations of the type "Data Element targets / sources Data Element" between Power BI Column assets and Column assets in Data Catalog. It also creates a data flow between data objects, which is visualized in a technical lineage.

> Note When you ingest Power BI, you automatically create a technical lineage for Power BI Column assets.

Stitching issues

To stitch assets in Data Catalog to data object collected by the lineage harvester, the Collibra Data Lineage server looks at the full path of the assets in Data Catalog and the full path of Power BI assets. If the full paths match, the Collibra Data Lineage automatically stitches them.

Usually, data objects that Collibra Data Lineage stitches to assets in Data Catalog have a yellow background in the technical lineage graph. However, the stitching results of BI sources, for example Power BI, currently have a gray background. This does not indicate that stitching failed. You can see which assets are stitched in the Stitching tab page.

> Tip You can use the Stitching tab page to easily find the full path of assets in Data Catalog and data objects that were collected by the Power BI harvester and the lineage harvester.

Schedule jobs

You can use scheduled jobs to run the Power BI harvester and lineage harvester at specific times automatically.

Since you need both the Power BI harvester and the lineage harvester to successfully ingest Power BI metadata in Data Catalog, we highly recommend that you schedule the Power BI harvester job before you schedule the lineage harvester job.

> Warning When you run the harvesters, Collibra Data Lineage creates all Power BI assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Power BI. As a consequence, all manually added data of those assets is lost.

Schedule Power BI harvester jobs

You can use the Windows Task Scheduler to make the Power BI harvester run scheduled jobs periodically. In a scheduled job, the Power BI harvester automatically uploads Power BI metadata to Collibra.

Scheduled jobs only work if you add the correct credentials to the Power BI configuration file or if you use a tool to automatically provide the credentials each time the Power BI harvester job is scheduled.

Schedule lineage harvester jobs

You can use Task Scheduler on Windows or Crontab on Mac and Linux to make the lineage harvester run scheduled jobs periodically. In a scheduled job, the lineage harvester uploads Power BI metadata to your Collibra Data Intelligence Cloud environment and Data Catalog automatically creates new Power BI assets and relations at specific times, dates or intervals. Collibra also creates a technical lineage for Power BI Column assets.

Warning Relations that were manually created between Power BI assets and other assets via a relation type in the Power BI operating model, are deleted after a refresh of the Power BI metadata.

Example You created a Power BI configuration file and added the required properties to the lineage harvester configuration file. You schedule the Power BI harvester job each Monday at 6 am and the lineage harvester job at 6 pm. As a result, your Power BI metadata is automatically refreshed on a weekly basis.

Harvesters upgrade

Each new Power BI harvester and lineage harvester adds features and enhancements to the previous version. We highly recommend that you always use the newest harvester available.

Upgrade to Power BI harvester 1.0.0.1 or newer and lineage harvester 1.3.0 or newer

The Power BI harvester 1.0.0.1 enables you to connect to a Collibra Data Lineage server, based on your geolocation and cloud provider.

You only have to follow this upgrade procedure when you upgrade from Power BI harvester 1.0.0.0 to Power BI harvester 1.0.0.1 and newer or if the server's geolocation or cloud provider changes.

Tip We highly recommend that you always use the newest Power BI harvester and lineage harvester.

Steps

1. If you have strict firewall rules, whitelist one of the following IP addresses, based on your Collibra geolocation and cloud provider:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)

    > Note IP address 15.222.200.199 is only available for Power BI harvester 1.0.0.2 and lineage harvester 1.3.1 and newer.

2. Download Power BI harvester 1.0.0.1 or newer, from the Collibra Downloads page.
3. Install the new Power BI harvester.
4. Migrate your Power BI connection information in your old configuration file to the configuration file in the new Power BI harvester folder.
5. Trigger the Power BI harvester to upload the Power BI metadata to the Collibra Data Lineage server with the new IP address:
    - Run the following command line if your configuration file is in its default location: `.\powerbi-harvester.bat`
    - Launch the path to the Power BI configuration file if you moved the configuration file to a different location: `.\bin\powerbi-harvester.exe .\config\powerbi-harvester.conf`

    > Note We highly recommend you to run the Power BI harvester via command line. This enables you to follow the metadata upload and see possible errors that may occur.

6. Download lineage harvester 1.3.0 or newer, from the Collibra Downloads page.
7. Install the new lineage harvester.
8. Migrate the data sources in your old configuration file to the configuration file in the new lineage harvester folder.

9. Run the lineage harvester with the `full-sync`command.
   » The lineage harvester uploads your data sources to the Collibra Data Lineage server with the new IP address.
10. Restart the DGC service in Collibra Console.

> Tip For more information about Power BI and the Power BI harvester, see Power BI.

What's next?

Collibra now synchronizes your Power BI assets and relations. You can also access the technical lineage via a Power BI Column asset page.

Power BI troubleshooting

It is possible that you encounter problems during the Power BI ingestion process.

> Note You can also encounter problems due to Power BI ingestion limitations.

The following table lists possible problems and offer a solution.

Problem	Description
You have made a mistake in the Power BI harvester configuration file or the lineage harvester configuration file.	Make sure to check all properties and values before you run the Power BI harvester. If any of the values of the required configuration properties are missing, invalid or incorrect, the Power BI harvester or lineage harvester fails with an error or the Power BI ingestion will be incorrect. The Power BI harvester and lineage harvester should have the same value in the `url` and `(source)Id` property.

Problem	Description
You don't have the correct Power BI permissions or not all prerequisites have been met before you start the Power BI integration process.	Make sure you have read and performed all prerequisites. The prerequisites are slightly different if you choose for username / password or service principal authentication.
The Power BI harvester failed to retrieve Power BI capacities and shows status code "Unauthorized".	This is a common mistake when you use the service principal authentication method. To solve this issue, make sure that you have enabled the **Allow service principals to use read-only Power BI admin APIs (preview)** option in the Power BI Admin portal,. Tip Do not confuse the **Allow service principals to use read-only Power BI admin APIs (preview)** option with the **Allow service principal to use Power BI APIs** option. You need to enable both options.
You have network or remote API issues.	Web services providing the API interfaces that the Power BI harvester uses may sometimes experience problems, or there may be problems with network access to these resources. If the Power BI harvester fails unexpectedly, check the following network resources and make sure they work properly: • Power BI REST API endpoints • XMLA endpoints • Technical lineage API Considering the nature of these remote resources, the cause of the problem can often be out of your control. Please wait until the issue is resolved or escalate the issue with the respective authority.

Problem	Description
You cannot retrieve information for individual Power BI dashboards or data sets.	To retrieve metadata of individual Power BI dashboards or data sets, you require permissions to access them. However, sometimes the Power BI dashboards and data sets are in a problematic state or you cannot reach them due to Power BI-related issues. When you execute the Power BI harvester, a summary of all encountered problems is printed. To reduce the number of problems, you can use the group filters in the Power BI configuration file to restrict the set of harvested Power BI workspaces. Depending on the type of issue, you may need to solve them one by one.
The Power BI harvester cannot retrieve certain workspaces in the `workspaceFilter` property.	Make sure the syntax in the `workspaceFilter` property in the configuration file is correct and you don't use the "IN" operator. > Note Currently, the "IN" operator is not supported. As a result, you cannot use "IN" to filter on specific Power BI workspaces in the `workspaceFilter` property in the Power BI configuration file. For more information, see the Power BI limitations.
The Power BI harvester failed to connect to the Microsoft API.	Usually, this is a timeout issue. We highly recommend that you increase the timeout duration. Use the following command line option to set the timeout duration: `--timeout <seconds>`. For example, if you want the Power BI harvester to wait 15 minutes for the connection, you can use `--timeout 900`.

Problem	Description
You get an error message related to Usage Metrics.	If you see errors related to Usage Metrics, you can ignore them, because they do not cause Power BI ingestion to fail. Usage Metrics are reports that are automatically created in Power BI, but they do not represent any Power BI assets or technical lineage information.
The technical lineage is missing or incomplete.	If the technical lineage is missing, you must add your Power BI workspaces to a dedicated capacity to allow the Power BI harvester to extract data from XMLA endpoints. Harvesting metadata via Power BI REST API does not require the dedicated capacity. As a result, the Power BI harvester can only reach limited Power BI metadata and won't create a technical lineage. If the technical lineage is incomplete, certain aspects of the Power BI ingestion job may not be supported. > Note You can only ingest new Power BI workspaces. This means that classic workspaces and My Workspace in Power BI is not supported. Also read the other limitations of the Power BI ingestion process to understand why technical lineage is missing or incomplete.
Some Power BI metadata is missing in Data Catalog.	Do the following: • Use new Power BI workspaces if you want a full ingestion. • Add your Power BI workspaces to a dedicated capacity to allow the Power BI harvester to extract data from XMLA endpoints. • Grant the Power BI workspaces the Contributor role in the Power BI security group.

Problem	Description
You have successfully ingested Power BI metadata, but calculated columns are not shown in the Technical lineage or in the browse tab pane.	Calculated columns are virtually the same as a non-calculated columns, with one exception: their values are calculated using DAX formulas and values from other columns. Collibra Data Lineage currently does not support internal transformations via DAX language, and any data objects derived via DAX are not shown in the technical lineage or in the browse tab pane. Currently, only M Query/Power Query expressions are supported.

Power BI ingestion tests

If you want to test the Power BI ingestion, we recommend that you use the `workspaceFilter` property in the Power BI configuration file to limit the Power BI ingestion to one or two Power BI workspaces.

For more information about the query options to filter Power BI workspaces, see the Microsoft documentation.

> Example If you want to limit the Power BI ingestion to one Power BI workspace with the name PowerBIWorkspace1, you can set the `workspaceFilter` value to "*Name eq 'PowerBIWorkspace1*".

Power BI harvester messages

When something goes wrong during the Power BI metadata harvesting process, the Power BI harvester logs show a message code that provides a link to more information. The message code indicates which part of the harvesting process failed or was skipped, and provides steps to resolve it.

> Tip Make sure that you understand the Power BI metadata harvesting process and the typical Power BI ingestion workflow.

Message code	Description
MSG-LIN-7000	This message is a reminder to follow all steps to ingest the Power BI metadata in Data Catalog. This message is always shown after the Power BI harvester successfully uploads the Power BI metadata to the Collibra Data Lineage server. Next, you have to create a lineage harvester configuration file and successfully run the lineage harvester to create Power BI assets and relations in Data Catalog.
MSG-LIN-7001	An unexpected problem occurred at the local machine. The error can be caused by an invalid path name, not enough storage space or other unexpected issues.
MSG-LIN-7002	There is a problem with the Power BI harvester configuration file or the source ID configuration file.Make sure that all information in the configuration file(s) and the path to the configuration file(s) is correct.
MSG-LIN-7003	The Power BI harvester could not retrieve tenant information, because the Microsoft API did not return a response that the Power BI harvester could process. To solve this problem, we recommend that you check your network settings and rerun the Power BI harvester. If the issue persists, please contact Collibra support or your customer success manager.

Message code	Description
MSG-LIN-7004	The Power BI harvester could not communicate with the Collibra Data Lineage server, likely because of one of the following scenarios: • The remote API did not return a response that the Power BI harvester could process. • The API call returned a 401 (Unauthorized) error because an invalid userKey token was used. To solve this problem, we recommend that you: • Ensure that the Power BI harvester is connecting to the correct Collibra Data Lineage server. • Check your network settings and rerun the Power BI harvester. If the issue persists, please contact Collibra support or your customer success manager.
MSG-LIN-7005	The Power BI harvester could not retrieve Power BI metadata, because the Power BI service did not return a response that the Power BI harvester could process. To solve this problem, we recommend that you check your network settings and rerun the Power BI harvester. If the issue persists, please contact Collibra support or your customer success manager. > Note If the error message indicates that the issue is an internal server error, the problem is caused by the Power BI REST API.

Message code	Description
MSG-LIN-7006	The Power BI harvester could not communicate with a remote server, because the server did not return a response within an expected time interval and, as a result, the Power BI harvester aborted the process. To solve this problem, we recommend that you do the following: • Check your network settings. • Check the amount of metadata that is processed. If it is a large amount, use the `--timeout` command line option to specify a longer timeout for remote API calls. If the problem persists or the remote server does not respond within a reasonable time period, create a support ticket or contact your customer success manager.
MSG-LIN-7007	This problem occurs when the Power BI service returns inconsistent data. As a result, the Power BI harvester cannot successfully process the data to create a consistent result data set. The Power BI harvester uses multiple API calls to retrieve Power BI metadata. If something in the Power BI service changed during the harvesting process, the metadata can be inconsistent. We recommend to run the Power BI harvester again. If the issue persists, create a support ticket or contact your customer success manager.
MSG-LIN-7008	The Power BI harvester cannot access XMLA endpoints for some Power BI dedicated capacities with harvested workspaces, because the capacities are currently not running. As a result, the Collibra Data Lineage server cannot create a technical lineage for these workspaces. To solve this problem, check if the Power BI workspace is part of a running dedicated capacity and you meet the necessary prerequisites to access and export it.

Message code	Description
MSG-LIN-7009	The Power BI authentication failed. This problem can be caused by an error in the Power BI credentials. To solve this problem, check your Power BI login credentials in the Power BI harvester configuration file or reenter them via command line.
MSG-LIN-7010	The connection between the Power BI harvester and the Collibra Data Lineage server failed. This problem can be caused by an error in the Collibra Data Intelligence Cloud credentials or Collibra Data Intelligence Cloud host address. To solve this problem, check your Collibra Data Intelligence Cloud credentials in the Power BI harvester configuration file or reenter them via command line.
MSG-LIN-7011	The Power BI harvester could not retrieve the tenant domain information. To solve this problem, check that you have the correct tenant domain ID in the Power BI harvester configuration file.

Message code	Description
MSG-LIN-7012	The Power BI API failed. This can be caused by an error in the syntax of the `workspaceFilter` field in the Power BI harvester configuration file. Important The `workspace filter` operations use OData syntax and are processed by the Power BI service, not the Power BI harvester. Examples of supported workspace filter operations: • `name eq 'Workspace1' or name eq 'Workspace2'` only harvests workspaces with the specified names. • `not endswith(name, 'Test')` only harvests workspaces whose names don't end in `Test`. • `tolower(capacityId) eq '01234567-89ab-cdef-0123-456789abcdef'` only harvests workspaces hosted on the specified dedicated capacity. • `reports/any(d:contains(d/name, 'Sales'))` only harvests workspaces with reports whose names contain `Sales`. If you do not want to filter on specific workspaces, leave the `workspaceFilter` field in the Power BI harvester configuration file empty. Tip For more information about the query options to filter Power BI workspaces, see the Microsoft documentation. We cannot guarantee that other group filter operations work correctly. For example, the `IN` operator is currently not supported.

Message code	Description
MSG-LIN-7013	You do not have the required permissions to harvest the Power BI metadata. Check that the user is a Power BI Administrator and that the Power BI application has all required permissions. To solve this problem, check that you have correctly registered your Power BI application in Microsoft Azure.
MSG-LIN-7014	You do not have the required permissions to harvest the Power BI metadata. Enable the **Allow service principals to use read-only Power BI admin APIs (preview)** option in the Power BI Admin Console. To solve this problem, check that you meet the prerequisites to use the service principal.
MSG-LIN-7015	You do not have the required permissions to harvest the Power BI metadata. Enable the **Allow service principal to use Power BI APIs** option in the Power BI Admin Console. To solve this problem, check that you meet the prerequisites to use the service principal.
MSG-LIN-7016	The harvested Power BI workspaces are not assigned to a dedicated capacity. As a result, Data Catalog cannot ingest details about tables and columns and technical lineage are not available. > Note You do not have to assign less important to a dedicated capacity, for example personal workspaces. However, if there are no workspaces on a dedicated capacity, the harvested Power BI metadata is very limited.

Message code	Description
MSG-LIN-7017	The Power BI harvester could not access XMLA endpoints for any Power BI workspaces to retrieve detailed information about data sets. As a result, technical lineage is be available. To solve this issue, check that you meet the prerequisites to access XMLA endpoints for all Power BI workspaces that you want to ingest in Data Catalog.
MSG-LIN-7018	Batch processing failed at Collibra server. The harvested batch was uploaded to a Collibra Data Lineage server, but the server could not process the batch. Review the error message that accompanies this error code. It might identify a problem that you can resolve, for example if you used an unsupported version of the harvester. If the error message does not identify the problem or if you're unable to resolve it on your own, create a support ticket or contact your customer success manager.

Working with SQL Server Reporting Services (SSRS) and Power BI Report Server (PBRS)

SQL Server Reporting Services and Power BI Report Server are server-based report generating applications created by Microsoft that helps you see and understand your data.

Power BI Report Server is included in the licensing of the SQL Server Enterprise Edition or as a free extension of Power BI premium. SQL Server Reporting Services and Power BI Report Server are closely related and both use the same API to communicate to the lineage harvester. As a result, Collibra created one operating model that contains data from both SQL Server Reporting Services and Power BI Report Server. Whether you use SQL Server Reporting Services, Power BI Report Server or both, you only need one integration in the lineage harvester configuration file.

> Note While SQL Server Reporting Services and Power BI Report Server use the same API, we can access less information from Power BI Report Server reports than from SQL Server Reporting Services data. As a result, we do not support stitching and lineage information for Power BI Report Server reports.

SQL Server Reporting Services and Power BI Report Server terminology

Before you ingest SQL Server Reporting Services and Power BI Report Server, read more about the terminology and how it maps with the Collibra Data Intelligence Cloud asset types.

Term	Description	Asset type in Collibra
Column	A column in an SQL Server Reporting Services Report Data Set.	SSRS Column
Data Set	A collection of data that is used to create an SQL Server Reporting Services Report.	SSRS Data Model
Folder	A collection of SQL Server Reporting Services and Power BI Report Server Reports and Data Sets.	SSRS Folder
KPI	A key performance indicator of SQL Server Reporting Services.	SSRS KPI
Mobile report	A detailed view of an SQL Server Reporting Services Data Set, with visualizations of findings and insights.	SSRS Report
Paginated report	A detailed view of an SQL Server Reporting Services Data Set, with visualizations of findings and insights.	SSRS Report

Term	Description	Asset type in Collibra
Parameter	A column that is part of an SQL Server Reporting Services Data Set and that is used in a KPI.	SSRS Parameter
Power BI Report Server report	A detailed view of a Power BI Data Model, with visualizations of findings and insights.	Power BI Report
SQL Server Reporting Services or Power BI Report Server server or tenant	A visual analytics platform for creating and storing SQL Server Reporting Services and Power BI Report Server Reports and Data Sets.	SSRS Server
Table	A table in an SQL Server Reporting Services Report Data Set.	SSRS Table

SQL Server Reporting Services and Power BI Report Server asset and domain types

The SQL Server Reporting Services and Power BI Report Server integration in Collibra Data Intelligence Cloud uses a specific subset of asset types and domain types. All of these come out of the box with your software.

The following table contains the asset types and domain types that are used for the SQL Server Reporting Services and Power BI Report Server integration. You can see the parent asset types in the breadcrumbs above each asset type.

Asset type	Description	Domain type
Business Asset ▸ Business Dimension ▸ BI Folder ▸ SSRS Folder	A collection of SQL Server Reporting Services and Power BI Report Server Reports and Data Sets.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ SSRS KPI	A key performance indicator of SQL Server Reporting Services.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ SSRS Report	A detailed view of an SQL Server Reporting Services Data Set, with visualizations of findings and insights.	BI Catalog
Data Asset ▸ Data Element ▸ Data Attribute ▸ BI Data Attribute ▸ SSRS Column	A column in an SQL Server Reporting Services Report Data Set.	BI Catalog

Asset type	Description	Domain type
Data Asset ▸ Data Element ▸ Report Attribute ▸ BI Report Attribute ▸ SSRS Parameter	A column that is part of an SQL Server Reporting Services Data Set and that is used in a KPI.	BI Catalog
Data Asset ▸ Data Set ▸ BI Data Set ▸ SSRS Data Model	A collection of data that is used to create an SQL Server Reporting Services Report.	BI Catalog
Data Asset ▸ Data Element ▸ Data Attribute ▸ BI Data Attribute ▸Power BI Table▸ SSRS Table	A table in an SQL Server Reporting Services Report Data Set.	BI Catalog
Technology Asset ▸ Server ▸ BI Server ▸ SSRS Server	A visual analytics platform for creating and storing SQL Server Reporting Services and Power BI Report Server Reports and Data Sets.	BI Catalog

Working with Looker

Looker is a business intelligence software that helps people see and understand their data.

For more information about Looker, see the Looker documentation.

> Note When you ingest Looker metadata, you automatically create a technical lineage for Looker.

Looker terminology

Before you ingest Looker, read more about the Looker terminology and how it maps with the Collibra Data Intelligence Cloud asset types.

> Note For more information, see the Looker documentation.

Looker term	Description	Asset type in Collibra
Dashboard	A collection of Looker tiles with metrics from one or more Looker Looks.	Looker Dashboard
Explore	A collection of data that is used to define Looker Dimensions and Measures.	Looker Data Set
Dimensions, Measures	An atomic unit of data that is used in a Looker Look or Looker Tile. It represents a column in a Looker Data Set.	Looker Data Set Column
Folder or Space	A container that stores Looker Looks, Dashboards and other folders.	Looker Folder
Look	A detailed view of a Looker Data Set, with visualizations of findings and insights.	Looker Look
Dimensions, Measures	An atomic unit of data that is used in a Looker Look or Looker Tile. It represents the actual use a Looker Data Set Column.	Looker Report Attribute
Query	A query that creates a simple report in a Looker Tile or Looker Look.	Looker Query
Looker instance	A platform to create Looker Dashboards and rich visualizations.	Looker Tenant
Tile or Dashboard element	An element that represents data on the Looker Dashboard.	Looker Tile

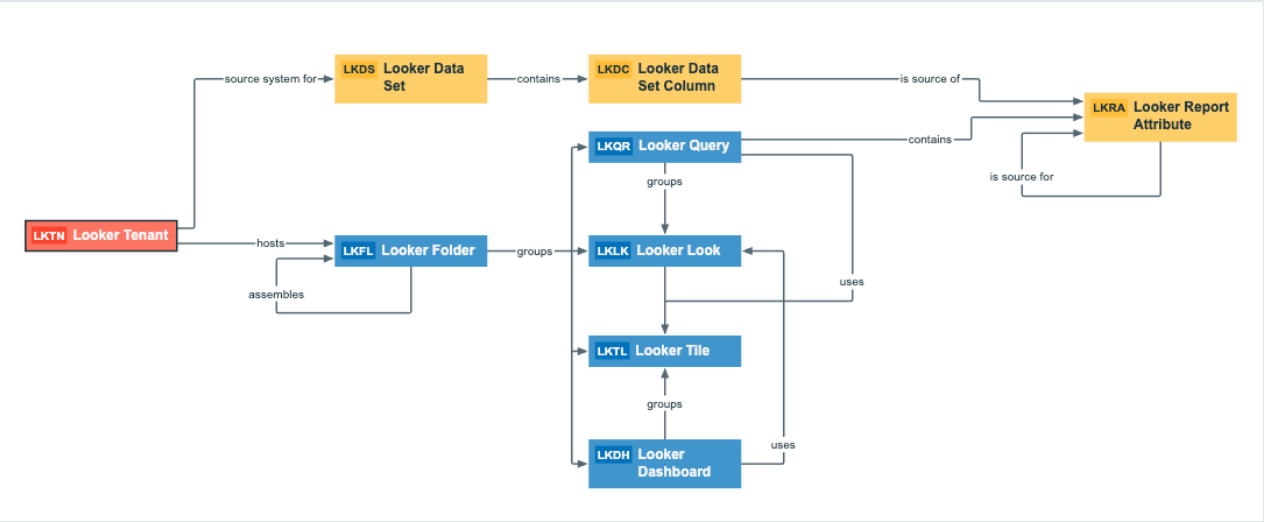
Looker operating model

The Looker scanner collects Looker metadata and sends it to the Collibra Data Lineage server. Collibra processes the metadata and creates new Looker assets and relations in Data Catalog. You can see them on the asset page overview or visualize them in a diagram or in a technical lineage.

> Note
> - The assets have the same names as their counterparts in Looker. Full names and Display names cannot be changed in Data Catalog.
> - Asset types are only created if you have all specific Looker and Data Catalog permissions.
> - All Looker asset types are created in the same domain.
> - Relations that were manually created between Looker assets and other assets via a relation type in the Looker operating model are deleted after a refresh of the Looker metadata.

Looker metadata overview

The following image shows the relations between Looker asset types.



Harvested metadata per asset type

The following table shows the harvested Looker metadata for each Looker asset type.

Asset type	Harvested Looker metadata in Data Catalog
Looker Dashboard	• Full name • Display name • Description • URL • Visit count • Favorites count • Document creation date • Document last accessed date • Business Dimension groups / is grouped into Report • Report uses / is used in Report • Report groups / is grouped into Report. • Report related to / impacted by Business Asset
Looker Data Set	• Full name • Display name • Description • Technology Asset source system for / source system Data Asset • Data Set contains / is part of Data Element
Looker Data Set Column	• Full name • Display name • Description • Data Set contains / is part of Data Element • Report Attribute sourced from / is source of Data Attribute
Looker Folder	• Full name • Display name • Document creation date • Server hosts / is hosted in Business Dimension • BI Folder assembles / is assembled in BI Folder • Business Dimension groups / is grouped into Report

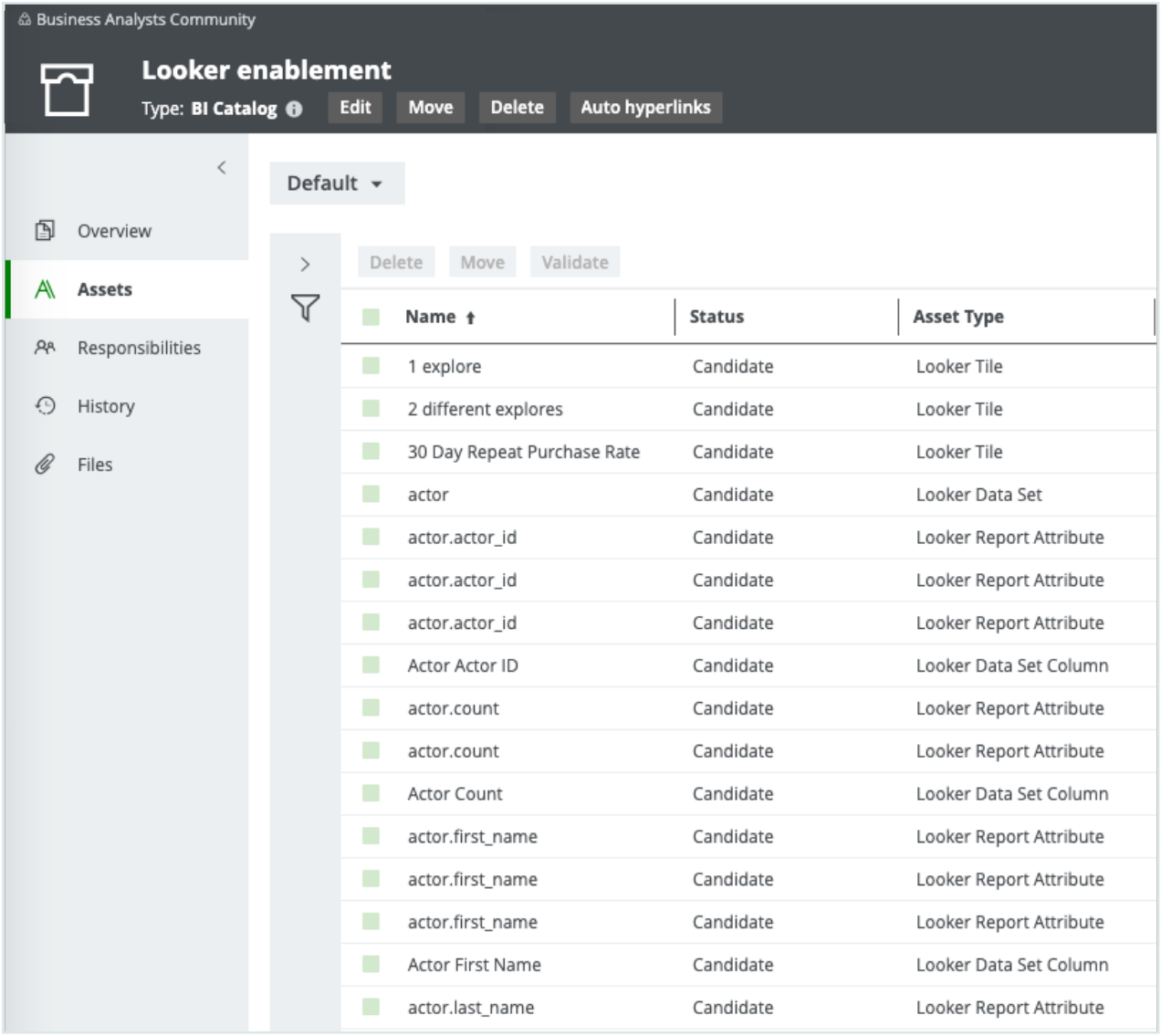
Asset type	Harvested Looker metadata in Data Catalog
Looker Look	• Full name • Display name • Description • URL • Visits count • Favorites count • Document creation date • Document modification date • Document last accessed date • Business Dimension groups / is grouped into Report • Report uses / is used in Report • Report groups / is grouped into Report
Looker Report Attribute	• Full name • Display name • Report Attribute contained in / contains Report • Report Attribute sourced from / is source of Data Attribute
Looker Query	• Full name • Display name • URL • Business Dimension groups / is grouped into Report • Report uses / is used in Report • Report Attribute contained in / contains Report
Looker Tenant	• Full name • Display name • Description • Server hosts / is hosted in Business Dimension • Technology Asset source system for / source system Data Asset

Asset type	Harvested Looker metadata in Data Catalog
Looker Tile	• Full name • Display name • Business Dimension groups / is grouped into Report • Report uses / is used in Report

> Note The metadata that is shown on the assets' pages depends on the asset type's assignment. As a result, you might not see all harvested metadata on the asset's page by default.

Example of ingested Looker metadata

The following image shows an example structure after Looker ingestion.



Looker asset and domain types

The Looker integration in Collibra Data Intelligence Cloud uses a specific subset of asset types and domain types. All of these come out of the box with your software.

The following table contains the asset and domain types that are used for the Looker integration. Above each asset type you can see the parent asset types in the breadcrumbs.

Asset type	Description	Domain type
Business Asset ▸ Business Dimension ▸ BI Folder ▸ Looker Folder	A container that stores Looker Looks, Dashboards and other folders.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Looker Dashboard	A collection of Looker tiles with metrics from one or more Looker Looks.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Looker Look	A detailed view of a Looker Data Set, with visualizations of findings and insights.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Looker Query	A query that creates a simple report in a Looker Tile or Looker Look.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ Looker Tile	An element that represents data on the Looker Dashboard.	BI Catalog

Asset type	Description	Domain type
Data Asset ▸ Data Element ▸ Data Attribute ▸ BI Data Attribute ▸ Looker Data Set Column	An atomic unit of data that is used in a Looker Look or Looker Tile. It represents a column in a Looker Data Set.	BI Catalog
Data Asset ▸ Data Element ▸ Report Attribute ▸ BI Report Attribute ▸ Looker Report Attrib-ute	An atomic unit of data that is used in a Looker Look or Looker Tile. It represents the actual use a Looker Data Set Column.	BI Catalog
Data Asset ▸ Data Set ▸ BI Data Set ▸ Looker Data Set	A collection of data that is used to define Looker Dimensions and Measures.	BI Catalog
Technology Asset ▸ Server ▸ BI Server ▸ Looker Tenant	A platform to create Looker Dashboards and rich visualizations.	BI Catalog

Overview Looker integration steps

The Looker integration enables you to harvest Looker metadata and create new Looker assets in Data Catalog. Collibra analyzes and processes the Looker metadata and presents it as specific asset types, retaining their original names.

> Tip To ingest Looker metadata in Data Catalog, you need to run the lineage harvester. The Looker ingestion workflow explains the role of the lineage harvester in the Looker ingestion process.

Steps

The table below shows the steps and prerequisites required to integrate Looker in Data Catalog.

Step	What?	Description	Prerequisites
1	Set up Looker authentication.	Before you start the Looker integration, you have to enable Collibra to access your Looker metadata.	• You have a Looker subscription.
2	Create a new domain.	Before you can ingest Looker metadata, you have to create a new domain or choose an existing domain to store the new Looker assets.	• You have a resource role with the following resource per-missions: ◦ Domain: Add

Step	What?	Description	Prerequisites
3	Download and install the lineage harvester and prepare a configuration file with Looker connection properties.	You use the lineage harvester to collect metadata from Looker and upload it to Collibra, where the metadata is scanned, processed and analyzed. When you download the lineage harvester, you can access the configuration file. You prepare a configuration file with Looker connection properties. > Note You need the lineage harvester 1.3.0 or newer to ingest Looker metadata into Data Catalog	• You have access to the lineage harvester 1.3.0 or newer.. • Your environment meets the system requirements to install and use the lineage harvester. • You have a global role that has the Manage all resources global permission. • You have a global role with the Catalog global permission, for example Catalog Author. • You have a global role with the Technical lineage global permission. • You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain: ◦ Asset: add ◦ Attribute: add ◦ Domain: add ◦ Attachment: add

Step	What?	Description	Prerequisites
4	Run the lineage harvester	You run the lineage harvester to start the ingestion process. Collibra creates new Looker assets in Data Catalog and imports relations between these assets. It also creates a technical lineage for Looker Look assets. You can create a lineage harvester job to schedule automatic Looker ingestion and synchronization.	• You have Collibra Data Intelligence Cloud 2020.12 or newer. • Your environment meets the system requirements to run the lineage harvester. • You have added Firewall rules so that the lineage harvester can connect to Collibra Data Lineage servers with the following IP addresses: ◦ 18.198.89.106 (techlin-aws-eu) ◦ 54.242.194.190 (techlin-aws-us) ◦ 15.222.200.199 (techlin-aws-ca) ◦ 35.205.146.124 (techlin-gcp-eu) ◦ 34.73.33.120 (techlin-gcp-us) ◦ 35.197.182.41 (techlin-gcp-au) ◦ 34.152.20.240 (techlin-gcp-ca) ◦ 51.105.241.132 (techlin-azure-eu) ◦ 20.102.44.39 (techlin-azure-us)

Step	What?	Description	Prerequisites
4	View the Looker assets and technical lineage	After the Looker metadata is ingested in Data Catalog, you can go to the domain where you ingested Looker and see the list of ingested Looker assets. You can go to a Looker Look asset page and click the Technical lineage lineage tab to view the technical lineage. Warning When you run the lineage harvester, Collibra Data Lineage creates all Looker assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Looker. As a consequence, all manually added data of those assets is lost.	• You have a global role with the Technical lineage global permission. • You have a global role with the Catalog global permission, for example Catalog Author.

Authentication

The Looker integration process uses a Looker API. To access the Looker metadata, the Looker API uses API3 credentials for authorization and access control.

Prerequisite

- You have the necessary permissions in Looker to see the Looker data.

Steps

1. Create a user with the Admin role.

   > Tip Only a user with a role that has the Admin permission set can create API3 credentials. Some Looker API calls also require a role that has the Admin permission set.

2. Create the API3 credentials.
3. Use the API3 credentials in the configuration file.

> Note API3 credentials are always linked to a Looker user account. As a result, calls to the API only return data that the user is allowed to see.

> Tip For more information, see the Looker documentation.

Prepare a domain for Looker ingestion

You can create a new domain for your Looker assets and use the domain ID in the lineage harvester configuration file. As a result, Collibra uses this domain to ingest all Looker assets during the Looker integration process.

Prerequisites

- You have a resource role with the Domain > Add resource permission.

Steps

1. In the main menu, click the **Create** (+) button.
   » The **Create** dialog box appears.

2. Click the **Organization** tab.
3. Click a domain type from the list.
   If you clicked the wrong domain type here, you can change it in the **Type** field in the next screen.
   » The **Create Domain** dialog box appears.
4. Enter the required information.

Field	Description
Type	The domain type of the domain you are creating. In this case, you need to select *BI Catalog*.
Community	The community under which the domain will be located.
Name	The name of the new domain.

5. Click **Create**.
6. Open your domain.
7. Copy the domain ID.

> Tip If you go to your domain, you can find the domain ID in the URL. The URL looks like: https://<yourcollibrainstance>/domain/**22258f64-40b6-4b16-9c08-c95f8ec0da26**?view=00000000-0000-0000-0000-000000040001. In this example, the domain ID is in bold.

8. Paste the domain ID in the lineage harvester configuration file.

> Warning When you run the lineage harvester, Collibra Data Lineage creates all Looker assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Looker. As a consequence, all manually added data of those assets is lost.

The lineage harvester setup for Looker

The lineage harvester is a software application that is needed to collect your Looker metadata and send it to the Collibra Data Lineage server, where the metadata is processed and new Looker assets and relations are created. Collibra Data Intelligence Cloud then import those assets and relations into Data Catalog.

For more information about the lineage harvester, read the Collibra Data Lineage section.

If you purchased Collibra Data Lineage, you have access to the lineage harvester on the Collibra downloads page.

For more information about the lineage harvester, read the Collibra Data Lineage section.

> Note You need the lineage harvester 1.3.0 or newer to ingest Looker metadata into Data Catalog

Lineage harvester system requirements

You need to meet the system requirements to be able to install and run the lineage harvester.

Software requirements

You need the following software requirements to install and run the lineage harvester.

Minimum software requirements

- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.

Recommended software requirements

- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.

Hardware requirements

You need to meet the hardware requirements to install and run the lineage harvester.

Minimum hardware requirements

You need the following minimum hardware requirements:

- 2 GB RAM
- 1 GB free disk space

Recommended hardware requirements

The minimum requirements are most likely insufficient for production environments. We recommend the following hardware requirements:

- 4 GB RAM
- 20 GB free disk space

Network requirements

You need the following minimum network requirements:

- Firewall rules so that the lineage harvester can connect to:
  - Your Collibra Data Intelligence Cloud instance version 2020.12 or newer.
  - All Collibra Data Lineage servers in your geographic location:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)

> Note The lineage harvester connects to different servers based on your geographic location and cloud provider. If your location or cloud provider changes, the lineage harvester rescans all your data sources. You have to whitelist all Collibra Data Lineage servers in your geographic location. In addition, we highly recommend that you always whitelist the techlin-aws-us server as a backup, in case the lineage harvester cannot connect to other Collibra Data Lineage servers.

> Note The lineage harvester uses port 443.

Install the lineage harvester for Looker integration

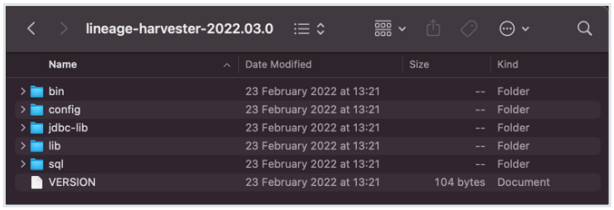
Before you can use the lineage harvester, you need to download it and install it. You can download the lineage harvester from the Collibra Community downloads page.

Prerequisites

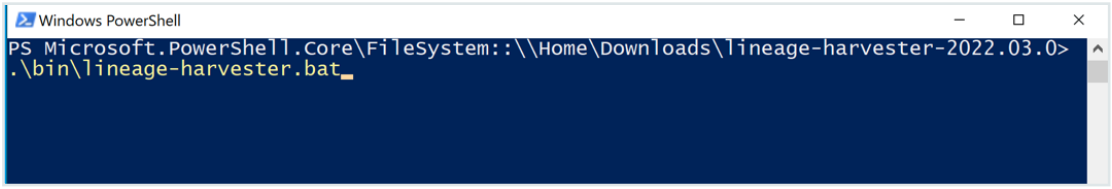
- You have purchased the Looker metadata connector and lineage feature.
- You have Collibra Data Intelligence Cloud 2020.12 or newer.
- You meet the minimum system requirements.
- Java Runtime Environment version 11 or newer or OpenJDK 11 or newer.
- You have added Firewall rules so that the lineage harvester can connect to:
  - the Collibra Data Lineage server with the following IP addresses:
    - 18.198.89.106 (techlin-aws-eu)
    - 54.242.194.190 (techlin-aws-us)
    - 15.222.200.199 (techlin-aws-ca)
    - 35.205.146.124 (techlin-gcp-eu)
    - 34.73.33.120 (techlin-gcp-us)
    - 35.197.182.41 (techlin-gcp-au)
    - 34.152.20.240 (techlin-gcp-ca)
    - 51.105.241.132 (techlin-azure-eu)
    - 20.102.44.39 (techlin-azure-us)
  - Collibra Data Intelligence Cloud 2020.12 or newer.

Steps

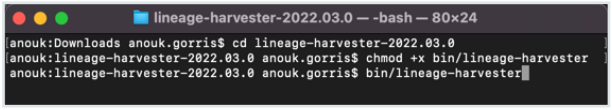
1. Download the lineage harvester version 1.3.0 or newer.
2. Unzip the archive.
   » You can now access the lineage harvester folder.



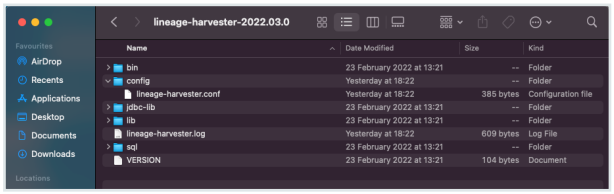
3. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



   - For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



   » An empty configuration file is created in the config folder.



   » The lineage harvester is installed automatically. You can check the installation by running `./bin/lineage-harvester --help`.

What's next?

You can now prepare the lineage harvester configuration file and run the lineage harvester to ingest Looker metadata into Data Catalog.

Prepare the lineage harvester configuration file for Looker

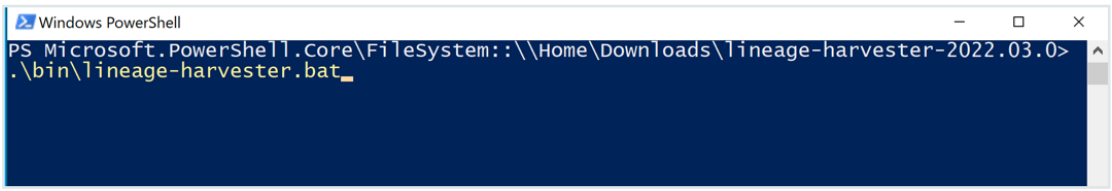
You have to prepare a configuration file before you run the lineage harvester. The lineage harvester collects your Looker metadata and sends it to the Collibra Data Lineage server, where it is processed and analyzed. Collibra Data Intelligence Cloud then imports the Looker assets and relations to Data Catalog.

Prerequisites

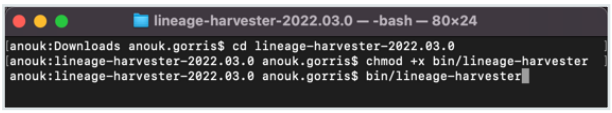
- You have Collibra Data Intelligence Cloud 2020.12 or newer.
- You have the lineage harvester 1.3.0 or newer.
- You have a global role that has the Manage all resources global permission.
- You have a global role with the Catalog global permission, for example Catalog Author.
- You have a global role with the Technical lineage global permission.
- You have created a BI Data Catalog domain in which you want to ingest the Looker assets.
- You have a resource role with the following resource permission on the community level in which you created the BI Data Catalog domain:
  - Asset: add
  - Attribute: add
  - Domain: add
  - Attachment: add
- You have downloaded the lineage harvester and you have the necessary system requirements to run it.

Steps

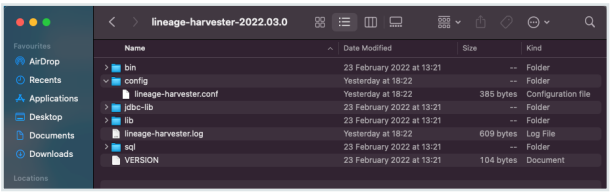
1. Run the following command line to start the lineage harvester:
   - Windows: `.\bin\lineage-harvester.bat`



- For other operating systems: `chmod +x bin/lineage-harvester` and then `bin/lineage-harvester`



» An empty configuration file is created in the config folder.



2. Open the lineage-harvester.conf file and enter the values for each property.

Properties	Description
general	This section describes the connection information between the lineage harvester and Data Catalog.
catalog	This section contains information that is necessary to connect to Data Catalog.
url	The URL of your Collibra Data Intelligence Cloud environment. Note You can only enter the public URL of your Collibra DGC environment. Other URLs will not be accepted.
username	The username that you use to sign in to Collibra.

Properties	Description
useCollibraSystemName	Indication whether you want to use the system or server name of a data source to match to the System asset you created when you prepared the physical data layer. This is useful when you have multiple databases with the same name. By default, the useCollibraSystemName property is set to `False`. If you want to use it, set it to `True`. ◦ If you keep the property set to `false`, the lineage harvester ignores the collibraSystemName property in the rest of the configuration file. ◦ If you set the useCollibraSystemName property to `true`, the lineage harvester reads the value in the collibraSystemName property in all sections of the configuration file and in the Looker <source ID> configuration file. Warning Unless you have multiple databases with the same name, we highly recommend that you keep the default value.
sources	This section contains all Looker connection properties.
collibraSystemName	This property is deprecated for Looker integration. The lineage harvester does not take into account any value that you enter here.

Properties	Description
id	The unique ID of your Looker metadata. For example, *my_looker*. Tip This value can be anything as long as it is unique and human readable. The ID identifies the batch of Looker metadata on the Collibra Data Lineage server.
type	The kind of data source. In this case, the value has to be *Looker*.
lookerUrl	The URL to your Looker API. Tip There are two ways to find the Looker API URL: ◦ In the API Host URL field in the Looker Admin menu. If this field is empty, you can use the default Looker API URL which you can find in the interactive API documentation. ◦ In the interactive API documentation URL. It is the part of the URL before `/api-docs/`.
clientId	The username you use to access the Looker API.
domainId	The unique ID of the domain in Collibra Data Intelligence Cloud in which you want to ingest the Looker assets.

3. Save the configuration file.

4. Start the lineage harvester again in the console and run the following command:
   - for Windows: `.\bin\lineage-harvester.bat full-sync`
   - for other operating systems: `./bin/lineage-harvester full-sync`
5. When prompted, enter the password or client secret to connect to your Collibra Data Intelligence Cloud and Looker environment.
   » The passwords are encrypted and stored in **/config/pwd.conf**.

Example

```
{
 "general": {
  "catalog": {
   "url": "https://<organization>.collibra.com",
   "userName": "<your-collibra-username>"
  },
  "useCollibraSystemName": false
 },
 "sources": {
  "collibraSystemName" : "",
  "id": "<looker-id>",
  "type": "Looker",
  "lookerUrl": "<https://<instance-name>.api.looker.com",
  "clientId": "<looker-api-user-name>",
  "clientSecret": "<looker-api-userkey",
  "domainId": "<domain-resource-id>"
 }
}
```

What's next?

The lineage harvester triggers Collibra to import Looker assets and their relations and create a technical lineage for Looker Look assets.

Currently, Looker assets are not yet stitched to other assets in Data Catalog.

If issues occur during the Looker ingestion process, check the Looker troubleshooting section to solve your problems.

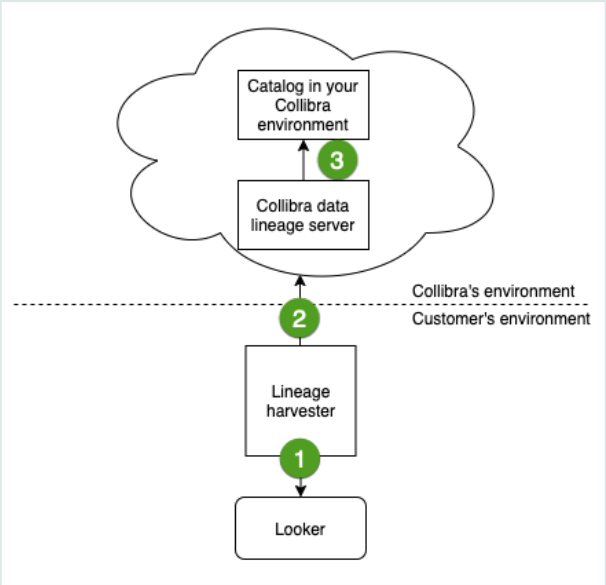
To refresh the Looker metadata, you can run the lineage harvester again or schedule jobs to run them automatically.

> Tip You can check the progress of the Looker ingestion in Activities. The results field indicates how many relations were imported into Data Catalog.

Looker ingestion workflow

You run the lineage harvester to start the Looker ingestion workflow. When you initiate Looker ingestion, each workflow component performs the following actions:

1. The lineage harvester:
   - Communicates with Looker.
   - Harvests the Looker metadata that will be ingested into Data Catalog.
   - Sends the Looker metadata to Collibra.
2. Collibra Data Intelligence Cloud:
   - Analyzes the Looker metadata.
   - Creates new assets and relations.
   - Imports new Looker assets and their relations in Data Catalog.
3. Data Catalog, via the Collibra Data Lineage server:
   - Shows new Looker assets.
   - Shows a technical lineage tab on Looker Look asset pages.



Collibra Data Lineage servers

A Collibra Data Lineage server processes and analyzes the harvested metadata and uploads it to Data Catalog. Collibra Data Lineage servers never process actual data.

Based on your geographical location and cloud provider, the lineage harvester sends metadata to one of the following Collibra Data Lineage servers:

- 18.198.89.106 (techlin-aws-eu)
- 54.242.194.190 (techlin-aws-us)
- 15.222.200.199 (techlin-aws-ca)
- 35.205.146.124 (techlin-gcp-eu)
- 34.73.33.120 (techlin-gcp-us)
- 35.197.182.41 (techlin-gcp-au)
- 34.152.20.240 (techlin-gcp-ca)
- 51.105.241.132 (techlin-azure-eu)
- 20.102.44.39 (techlin-azure-us)

> Important You have to whitelist all Collibra Data Lineage servers in your geographic location. For example, if your data is located in Europe, you have to whitelist the following Collibra Data Lineage servers: techlin-aws-eu and techlin-gcp-eu. In addition, we highly recommend that you always whitelist the techlin-aws-us Collibra Data Lineage server as a backup, in case the lineage harvester cannot connect to other Collibra Data Lineage servers.

Schedule Looker ingestion jobs

You can use Task Scheduler on Windows or Crontab on Mac and Linux to make the lineage harvester run scheduled jobs. In a scheduled job, the lineage harvester uploads the Looker data source information to Collibra.

Collibra automatically creates new assets and relations of the type "Data Element targets / sources Data Element" at specific times, dates or intervals, using the information in your configuration file.

> Warning When you run the lineage harvester, Collibra Data Lineage creates all Looker assets in the same Data Catalog BI domain. We highly recommend that you do not move these assets to another domain. If you move assets to another domain, they will be deleted and recreated in the initial Data Catalog BI domain when you synchronize Looker. As a consequence, all manually added data of those assets is lost.

Warning Relations that were manually created between Looker assets and other assets via a relation type in the Looker operating model, are deleted after a refresh of the Looker metadata.

Example You created a configuration file with connection information to your Looker environment. You schedule the lineage harvester job to run each Sunday at 23:00. As a result, your Looker metadata is automatically refreshed on a weekly basis.

Looker business logic

Looker business users usually work with Looker dashboards and Looker looks to make business decisions. Collibra's Looker connector and lineage feature, offers business users several advantages:

- Easily find certified Looker content.
- Shop for Looker Looks.
- Find where content is stored in Looker.
- Get information about a Looker Look and other Looker report details in a single location.

Note Due to limitations of the Looker REST API, Data Catalog cannot stitch Looker assets and corresponding assets in Data Catalog. The Looker REST API does not provide transformations in Looker that are needed for stitching.

Looker asset pages

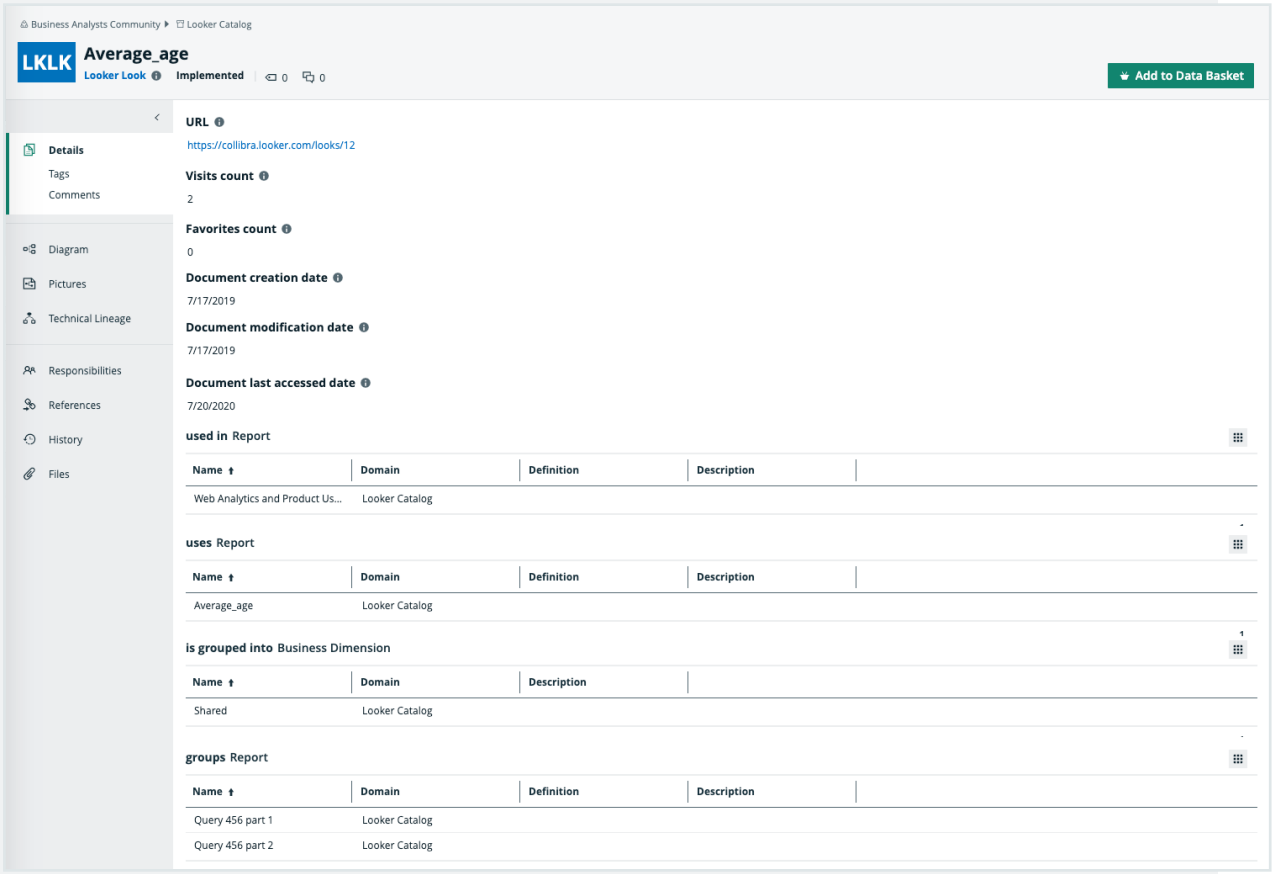
Depending on the Looker asset type, the asset page shows different information ingested from Looker. You can find a specific Looker asset page using Data Catalog search or via the Data Catalog BI domain in which you ingested the Looker metadata.

Details

An asset page contains attributes and relations to other assets. This information is synchronized from Looker. However, you can add additional characteristics, tags or comments.

If you want to use a Looker Look, you can add it to the Data Basket and check it out.

Example The following Looker Look asset shows in which Looker Folder it is stored, in which Looker Dashboard it is shown, which Looker Tiles it uses and which Looker Queries it groups. This asset has a number of attributes that give more information about the Looker Look.

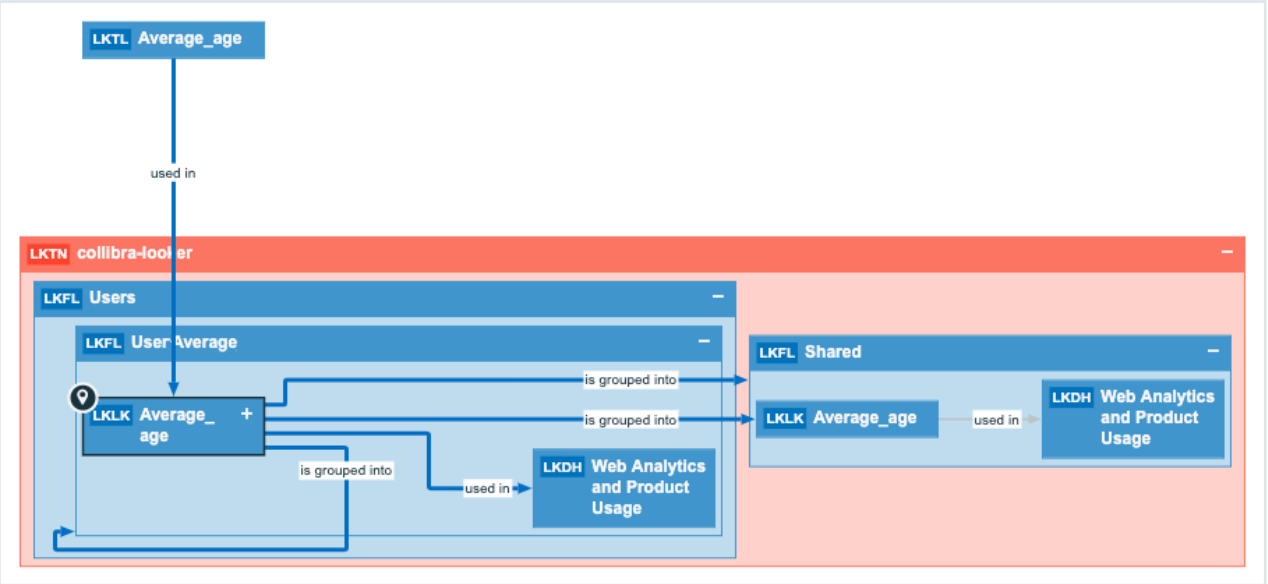


Business diagrams

The business diagram is a feature to show and interact with many assets and relations in an easy-to-read diagram. The business diagram helps you to quickly see to which other assets a specific asset is related. As such, the diagram can show a high-level presentation

of a Looker Look. This enables you to see how the Looker Look relates to other Looker assets.

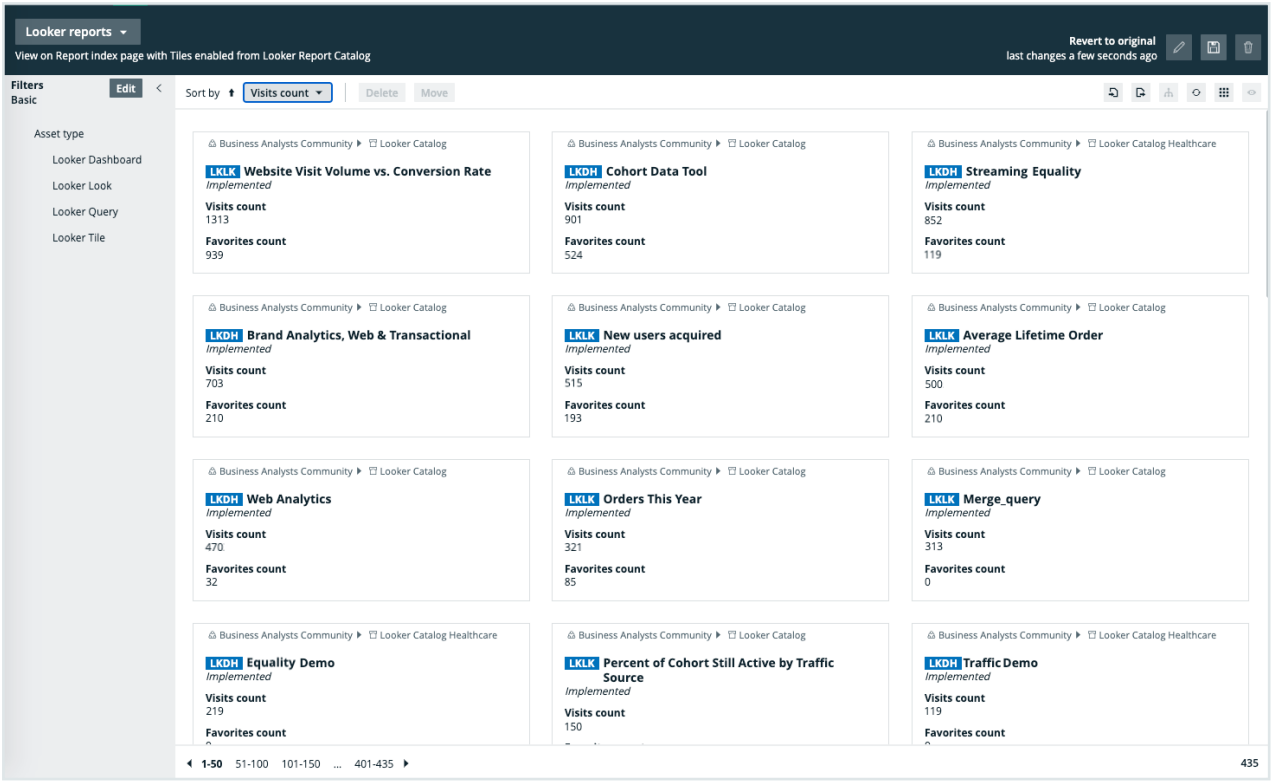
Example The following business diagram shows the *Average_age* Looker Look, which is stored in the *UserAverage* Looker Folder, but is also grouped into the *Shared* Looker Folder.



Report views

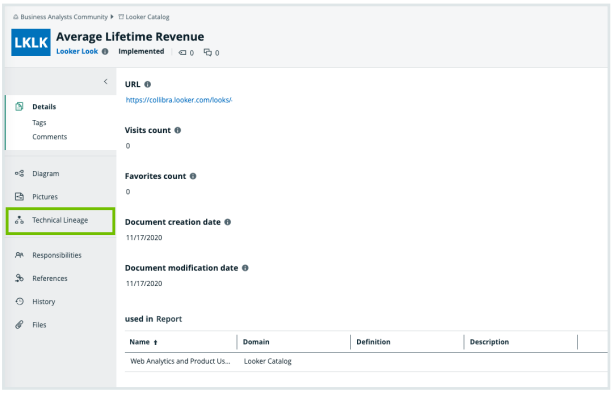
The Looker connector and lineage feature enables you to find all ingested Looker Look, Looker Dashboard, Looker Tile and Looker Query asset types in a single location.

In the **Reports** tab page in Data Catalog you can see an overview of all Report assets and their children. Optionally, you can create a view with a filter to only show Looker assets. This is useful if you quickly want to see all reports or if you want find specific reports for example certified reports or reports that are visited the most.



Technical lineage for Looker

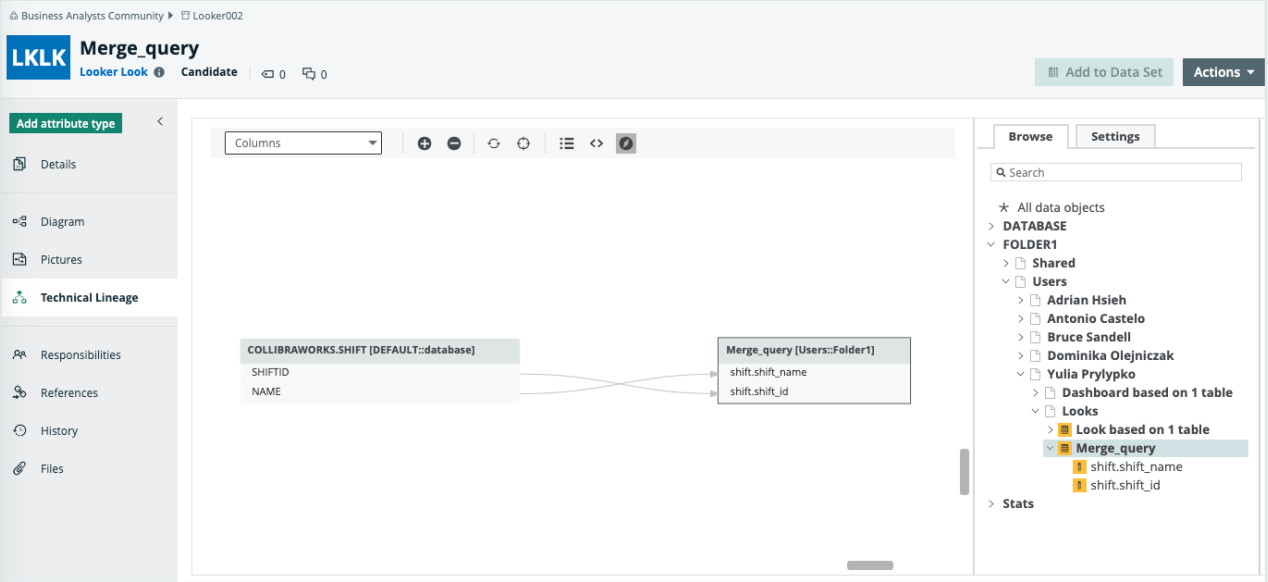
When you ingest Looker metadata, you automatically create a technical lineage for Looker Look assets. If you have the right permissions to view the technical lineage, you can go to a Looker Look asset page and click the Technical lineage tab, which allows you to access the technical lineage.



> Note Due to the limitations of the Looker REST API, we cannot stitch Looker assets and corresponding assets in Data Catalog. The Looker REST API does not provide transformations in Looker that are needed for stitching. As a result, the technical lineage only shows Looker metadata as it exists on the Collibra Data Lineage server and not as assets in Data Catalog.

Example

The following technical lineage graph shows the technical lineage of Looker objects.



Troubleshooting

If the connection between the lineage harvester and your Looker instance fails, you must try to sign in to Looker as an Admin user on the same machine that runs the lineage harvester. Open the interactive API documentation. If you are not able to open the API page, try one of the following:

- Check that you have network access to the API URL.
- Check that you have the correct credentials to sign in to the interactive API documentation. If necessary, create new API3 keys and try again. If you are now able to access the interactive API documentation, use the new Client ID and Client Secret in the configuration file.

- Sign in to the interactive API documentation with your API3 credentials and test the API calls. If your test is successful, compare API URL in the Request URL section to the `lookerUrl` value in the configuration file.

> Tip There are two ways to find the Looker API URL:
>
> - In the API Host URL field in the Looker Admin menu. If this field is empty, you can use the default Looker API URL which you can find in the interactive API documentation.
> - In the interactive API documentation URL. It is the part of the URL before `/api-docs/`.

Working with MicroStrategy

MicroStrategy Intelligence Server is business intelligence software that connects to data sources to create and store layers of objects in the MicroStrategy metadata.

For more information about MicroStrategy, see the MicroStrategy documentation.

> Note
> - MicroStrategy is available as a harvester integration in beta. If you want to test the MicroStrategy ingestion, you can create a support ticket to request testing guidelines.
> - You can access any local or remote PostgreSQL database. The MicroStrategy Intelligence Server has an embedded PostgreSQL repository, as its default repository. For complete information on the default, embedded repository, see the MicroStrategy repository documentation.

MicroStrategy terminology

Before you ingest MicroStrategy, read more about the MicroStrategy terminology and how it maps with the Collibra Data Intelligence Cloud asset types.

MicroStrategy term	Description	Asset type in Collibra
Attribute	A detailed view of a MicroStrategy visualization, with findings and insights.	MicroStrategy Report Attribute

MicroStrategy term	Description	Asset type in Collibra
Column	A column in a MicroStrategy data model.	MicroStrategy Column
Dataset	A collection of data that is used to create MicroStrategy reports.	MicroStrategy Data Model
Dossier	A collection of MicroStrategy chapters and pages.	MicroStrategy Dossier
Folder	A collection of MicroStrategy reports and data models.	MicroStrategy Folder
Project	A collection of MicroStrategy visualizations, report attributes and tables.	MicroStrategy Project
Report	A detailed view of a MicroStrategy data model, with visualizations of findings and insights.	MicroStrategy Report
Server	A visual analytics platform for creating and storing MicroStrategy reports and data models.	MicroStrategy Server

MicroStrategy asset and domain types

The MicroStrategy integration in Collibra Data Intelligence Cloud uses a specific subset of asset types and domain types. All of these come out of the box with your software.

The following table contains the asset and domain types that are used for the MicroStrategy integration. Above each asset type you can see the parent asset types in the breadcrumbs.

Asset type	Description	Domain type
Business Asset ▸ Business Dimension ▸ BI Folder ▸ MicroStrategy Folder	A collection of MicroStrategy reports and data models.	BI Catalog
Business Asset ▸ Business Dimension ▸ BI Folder ▸ MicroStrategy Project	A collection of MicroStrategy visualizations, report attributes and tables.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ MicroStrategy Dossier	A collection of MicroStrategy chapters and pages.	BI Catalog
Business Asset ▸ Report ▸ BI Report ▸ MicroStrategy Report	A detailed view of a MicroStrategy data model, with visualizations of findings and insights.	BI Catalog
Data Asset ▸ Data Element ▸ Data Attribute ▸ BI Data Attribute ▸ MicroStrategy Column	A column in a MicroStrategy data model.	BI Catalog

Asset type	Description	Domain type
Data Asset ▸ Data Element ▸ Report Attribute ▸ BI Report Attribute ▸ MicroStrategy Report Attribute	A detailed view of a MicroStrategy visualization, with findings and insights.	BI Catalog
Data Asset ▸ Data Structure ▸ Data Model ▸ BI Data Model ▸ MicroStrategy Data Model	A collection of data that is used to create MicroStrategy reports.	BI Catalog
Technology Asset ▸ Server ▸ BI Server ▸ MicroStrategy Server	A visual analytics platform for creating and storing MicroStrategy reports and data models.	BI Catalog