MANTA Flow Installation and Usage Manual

- 1 Introduction 5
- 2 Setup Process 5
  - 2.1 Prerequisites 5
    - 2.1.1 Hardware Requirements 5
      - 2.1.1.1 Minimum Configuration 5
      - 2.1.1.2 Recommended Configuration 5
    - 2.1.2 Software Requirements 5
      - 2.1.2.1 Known Issues 5
    - 2.1.3 User Access Requirements 6
    - 2.1.4 Java Installation Instructions 6
    - 2.1.5 MANTA Installation Files 6
  - 2.2 Installation 6
  - 2.3 Upgrade to a Newer Version 7
  - 2.4 Instructions for Versions R36 and Newer 7
  - 2.5 Instructions for Versions R35.1 and Older 8
    - 2.5.1 Upgrade the MANTA Updater Application 8
    - 2.5.2 Upgrade MANTA Flow to a Newer Version 9
- 3 Architecture 9
  - 3.1 MANTA Flow Components 9
  - 3.2 MANTA Flow High-Level Architecture 9
  - 3.3 MANTA Flow CLI and MANTA Flow Server Detailed Architecture 10
  - 3.4 MANTA Flow Deployment Diagram: Installer 10
  - 3.5 MANTA Flow Deployment Diagram: Containerized 11
  - 3.6 MANTA Flow Security Architecture 11
  - 3.7 Lineage Scanning Architecture 11
  - 3.8 Overview of the MANTA Flow Scanning Process 11
    - 3.8.1 What Is a Scan? 11
    - 3.8.2 Phases of a Scan 12
    - 3.8.3 What Is a Revision? 12
    - 3.8.4 What Is an Export? 12
  - 3.9 Details Regarding Individual Phases 12
    - 3.9.1 Extraction Phase 12
    - 3.9.2 Analytic Phase 14
    - 3.9.3 Export Phase 16
- 4 Lineage Scanner Configuration 17
  - 4.1 Databases 17
    - 4.1.1 BigQuery 17
    - 4.1.2 DB2 17
      - 4.1.2.1 IBM DB2 for Linux, UNIX, and Windows 17
      - 4.1.2.2 IBM DB2 for z/OS 19
    - 4.1.3 Hive 19
      - 4.1.3.1 Environment with Direct Access to Hive Server 2 via JDBC 20
    - 4.1.4 Microsoft SQL Server, Azure SQL Database, Azure SQL Data Warehouse, SAP (Sybase) ASE, Parallel Data Warehouse / Analytic Data Platform, and Amazon RDS for SQL Server 20
    - 4.1.5 Netezza 22
    - 4.1.6 Oracle 22
    - 4.1.7 PostgreSQL, Greenplum, Yellowbrick, Redshift, Amazon RDS, or Amazon Aurora for PostgreSQL 23
    - 4.1.8 Snowflake 23
      - 4.1.8.1 Supported Features 24
      - 4.1.8.2 Examples of Unsupported Features 24
      - 4.1.8.3 Known Issues 24
    - 4.1.9 Teradata 24
  - 4.2 ETL Tools 26
    - 4.2.1 IBM DataStage 26
      - 4.2.1.1 Supported Features 26
      - 4.2.1.2 Wise Column Connection 29
      - 4.2.1.3 Examples of Unsupported Features 29
    - 4.2.2 Informatica PowerCenter 29
      - 4.2.2.1 PowerCenter Scanner Overview 29
        - 4.2.2.1.1 Supported Features 29
        - 4.2.2.1.2 Examples of Unsupported Features 30
    - 4.2.3 Microsoft SSIS 31

– 4.2.3.1 Examples of Unsupported Features ... 31
› 4.2.4 Oracle Data Integrator ... 31
› 4.2.5 Talend ... 32
› 4.2.6 Exporting jobs ... 32
› 4.2.7 Building jobs ... 32
› 4.2.8 Pig Latin ... 32
› 4.2.9 Sqoop ... 33
› 4.2.10 StreamSets ... 33
– 4.2.10.1 Supported Features ... 33
– 4.2.10.1.1 JSP 2.0 Expression Language (EL) ... 35
– 4.2.10.2 Unsupported Features ... 35
– 4.2.10.3 Extraction of Jobs ... 35
› 4.2.11 Kafka ... 35
– 4.2.11.1 Supported Features ... 35
– 4.2.11.2 Unsupported Features ... 36
› 4.2.12 Schema Registry Integration Requirements ... 36
› 4.2.13 Manually Provided Schema Integration Requirements ... 36
– 4.2.13.1 Field Overview ... 37
› 4.2.14 Azure Data Factory ... 38
› 4.2.15 Supported Features ... 38
– 4.2.15.1 Legend ... 38
– 4.2.15.2 Supported Resources ... 38
– 4.2.15.3 Data Flow Transformations ... 38
› 4.2.16 Examples of Unsupported Features ... 39
› 4.2.17 Fivetran ... 39
– 4.2.17.1 Fivetran Scanner Overview ... 40
– 4.2.17.1.1 Supported Features ... 40
– 4.2.17.1.1.1 Legend ... 41
– 4.3 BI and Reporting Tools ... 41
› 4.3.1 IBM Cognos ... 41
– 4.3.1.1 Cognos SDK Libraries ... 42
– 4.3.1.2 Cognos Scanner Overview ... 42
– 4.3.1.2.1 Supported Features ... 42
– 4.3.1.2.2 Examples of Unsupported Features ... 43
› 4.3.2 Microsoft Excel ... 43
› 4.3.3 Microsoft Power BI ... 43
› 4.3.4 Azure Extraction Mode ... 43
– 4.3.4.1 Scanner API ... 44
› 4.3.5 Local Extraction Mode ... 44
› 4.3.6 Supported Features ... 44
› 4.3.7 Examples of Unsupported Features ... 45
› 4.3.8 Feedback ... 45
› 4.3.9 Microsoft SSAS ... 45
› 4.3.10 SSAS Scanner Overview ... 45
› 4.3.11 Input Example ... 46
› 4.3.12 Supported Features ... 46
› 4.3.13 Examples of Unsupported Features ... 46
› 4.3.14 Microsoft SSRS ... 46
– 4.3.14.1 REST Extraction Mode ... 46
– 4.3.14.2 DATABASE Extraction Mode ... 47
› 4.3.15 OBIEE ... 47
– 4.3.15.1 Manual Extraction ... 48
– 4.3.15.2 OBIEE Scanner Overview ... 48
– 4.3.15.3 Supported Features ... 48
– 4.3.15.4 Examples of Unsupported Features ... 49
› 4.3.16 Tableau ... 49
– 4.3.16.1 Providing Input Data Manually ... 49
– 4.3.16.2 Permissions ... 50
– 4.3.16.3 Supported Features ... 50
– 4.3.16.4 Known Limitations ... 50
› 4.3.17 SAP BusinessObjects ... 51
– 4.3.17.1 SAP BO Scanner Overview ... 52
– 4.3.17.1.1 Extractor Usage ... 52
– 4.3.17.1.1.1 Detailed Explanation of the Extraction Scenarios Mentioned Above ... 52
– 4.3.17.1.2 Supported Features ... 53
– 4.3.17.1.3 Examples of Unsupported Features ... 53
› 4.3.18 Qlik Sense ... 53

- 4.3.18.1 Prerequisites 54
- 4.3.18.2 Qlik Sense Scanner Overview 56
- 4.3.18.3 Supported Features 56
- 4.3.18.4 Extraction 56
- 4.3.18.5 Data Flow Analysis 56
  - 4.3.18.5.1 Load Script Statement Analysis 56
- 4.3.18.6 Unsupported Features 56
  - 4.3.18.6.1 Extraction 56
  - 4.3.18.6.2 Data Flow Analysis 57
  - 4.3.18.6.3 Load Script Statement Analysis 57
- 4.3.19 SAS 57
  - 4.3.19.1 Supported Features 57
- 4.4 Data Modeling Tools 58
  - 4.4.1 E/R Studio 58
  - 4.4.2 Erwin 58
  - 4.4.3 SAP PowerDesigner 58
- 4.5 Programming Languages 59
  - 4.5.1 Cobol/JCL 59
  - 4.5.2 Java 59
  - 4.5.3 C# 60
  - 4.5.4 Python 61
    - 4.5.4.1 Prerequisites 61
    - 4.5.4.2 Supported Features 61
    - 4.5.4.3 Extraction 61
    - 4.5.4.4 Data Flow Analysis 61
    - 4.5.4.5 Unsupported Features 62
- 4.6 Other 62
  - 4.6.1 Filesystem 62
  - 4.6.2 File Path Mapping Configuration 62
  - 4.6.3 Mapping Rules 63
  - 4.6.4 Source Technology Values 63
  - 4.6.5 Examples 64
- 5 MANTA Server Administration 65
  - 5.1 HTTPS Configuration 65
  - 5.2 SSO Configuration 66
  - 5.3 Encoding 66
  - 5.4 Authorization 67
    - 5.4.1 This article describes the overall concepts of authentication and authorization in MANTA Flow Server. There are two guides that explain how to configure the authentication. Choose the right guide according to the version of MANTA you are using. 67
    - 5.4.2 User Roles 67
    - 5.4.3 Disabling Security for Specific Parts 67
    - 5.4.4 Access Rights for the Metadata Repository 67
      - 5.4.4.1 Enabling and Disabling Access Rights 67
      - 5.4.4.2 Defining Access Rights 67
      - 5.4.4.3 Repository View Configuration 68
      - 5.4.4.4 Assigning Repository Views to User Roles 68
      - 5.4.4.5 Applying the Changes 68
      - 5.4.4.6 Repository Object Permission Evaluation 68
      - 5.4.4.7 Example Configuration 68
  - 5.5 Repository 69
  - 5.6 Back Up and Restore 69
    - 5.6.1 Physical Copy 69
    - 5.6.2 Dump 69
      - 5.6.2.1 REST API 70
  - 5.7 Maintenance Links 70
  - 5.8 Location of Persistent Files 70
  - 5.9 Advanced Configuration (Titan Database up to R35) 71
    - 5.9.1 Session Policy 71
      - 5.9.1.1 Cookies 71
  - 5.10 Version Control System 72
    - 5.10.1 Revision System 72
  - 5.11 Logging 72
  - 5.12 Usage Statistics 72
  - 5.13 Module Specific Configuration 75
    - 5.13.1 Merger 75
- 6 Usage 75
  - 6.1 Preparing Input Scripts 75

– 6.2 Default Metadata Input 75
– 6.3 API Metadata Input 75
– 6.4 Technology Directory Contents 75
› 6.4.1 Oracle PL/SQL 75
› 6.4.2 Teradata 76
› 6.4.3 SQL Server and T/SQL 76
› 6.4.4 HiveQL 76
› 6.4.5 Netezza and NZPLSQL 76
› 6.4.6 PostgreSQL, Greenplum, Redshift, Yellowbrick, Amazon RDS, or Amazon Aurora for PostgreSQL and PL/pgSQL 76
› 6.4.7 DB2 PL/SQL 76
› 6.4.8 Informatica PowerCenter 77
› 6.4.9 SSIS 77
› 6.4.10 SSAS 78
› 6.4.11 Talend 78
› 6.4.12 DataStage 78
› 6.4.13 Pig Latin 78
› 6.4.14 Sqoop 78
› 6.4.15 Excel 78
› 6.4.16 ER/Studio 78
› 6.4.17 PowerDesigner 78
› 6.4.18 Cobol/JCL 79
› 6.4.19 ODI 79
› 6.4.20 Snowflake 79
› 6.4.21 StreamSets 79
› 6.4.22 BigQuery 79
› 6.4.23 SAP HANA 79
› 6.4.24 Python 80
› 6.4.25 Qlik Sense 80
› 6.4.26 Kafka 81
› 6.4.27 Fivetran 81
– 6.4.27.1 Destination Metadata Files 81
– 6.4.27.1.1 API Authorization 82
– 6.4.27.1.2 The Postman Collection with Endpoints for Easier Extraction 82
– 6.4.27.2 SQL Transformation Project 83
– 6.4.27.2.1 Basic Transformation Project 83
– 6.4.27.2.2 Transformations for dbt Core 84
– 6.4.27.2.2.1 Schema Names 84
› 6.4.28 Azure Data Factory 85
– 6.4.28.1 Export from Git via Command Line 85
– 6.4.28.2 Manual Download from GitHub 86
– 6.4.28.3 Manual Download from ADF 86
› 6.4.29 Unqualified Object References 86
› 6.4.30 87
– 6.5 Running a Data Flow Analysis 87
– 6.6 Reading Outputs 88
› 6.6.1 Data Flow Viewer 88
› 6.6.2 CSV Files 88
– 6.7 Scheduling a Data Flow Analysis Run 88
› 6.7.1 Considerations When Scheduling an Analysis Run 88
– 6.8 Export User Documentation 88
› 7 Export Execution 88
› 8 Metadata Format 89
› 9 Metadata Analysis 89
– 9.1 Finding a Fully-Qualified Object 89
– 9.2 Listing All Object Attributes 90
– 9.3 Finding All Direct Flows from an Object (One Step) 90
– 9.4 Finding All Direct Flows from an Object (Recursive) 90
– 9.5 Repository API 91
– 9.6 Introduction 91
– 9.7 Authentication and Authorization 91
– 9.8 Revision Differences 92
– 9.9 Prerequisites 92
– 9.10 Usage 92
› 9.10.1 Restrictions 92
› 9.10.2 Examples 92
› 9.10.3 Usage with MANTA Flow Client 93
– 9.11 Difference Report Format 93
› 9.11.1 Example of a Report 93

Introduction

This document provides information about the installation, configuration, and usage of MANTA Flow when using an automated installer that guides you through the installation process. More complete technical documentation for advanced configuration is also available.

Setup Process

This section explains how to install and upgrade MANTA Flow.

Prerequisites

This section summarizes all the prerequisites that must be fulfilled before starting MANTA.The following technical requirements must be met to use MANTA Software. Note that these requirements do *not* take into account any third-party products, services, or integrations that you may decide to run. Please refer to the documentation of any such third-party software for its technical requirements.

Hardware Requirements

A dedicated machine for MANTA is recommended to avoid resource contention and limit MANTA's access to other data and applications for security reasons.

Minimum Configuration

CPU: 4 cores at 2.5 GHz

RAM: 16 GB

HDD: 2 GB for MANTA installation + 50 GB of space for metadata; SSD, minimum of 1500 IOPS

Recommended Configuration

CPU: 8 cores at 3 GHz

RAM: 24–256 GB

HDD: 2 GB for MANTA installation + between 200 GB and 1 TB of space for metadata; SSD, minimum of 2000 IOPS

Software Requirements

OS: Windows 7 / Server 2008 or newer, Linux or Solaris, Mac (without installer)

Java: 64-bit JRE/JDK (Java Runtime Environment / Java Development Environment) is a prerequisite for MANTA. (See the compatibility matrix below.) The recommended version is Java 11. JRE/JDK is NOT part of the MANTA package. It is the sole responsibility of the customer to obtain JRE/JDK. The distributions tested by MANTA are OpenJDK and Oracle. Although it is not guaranteed that other distributions of JRE/JDK will work with MANTA, distributions based on the two previously mentioned are likely to work fine. For example, IBM JRE/JDK is not supported. Each distributor has its own license conditions, and it is up to the customer to fulfill those conditions. MANTA is not in any way responsible for the customer´s compliance with the required conditions.

MANTA/Java	8	9	10	11	12	13	14	15	16	17	18
R29-R30											
R31-R32											
R33											
R34-R35											
R36											
R37											

Known Issues

- For Java 8:
  - Only Java 8u152 or newer is supported.
  - Java 8u242 is not supported due to issues with handling certificates. (The defect is not present in earlier updates and is fixed in later updates.)
- Java 11.0.14 is not supported due to a defect introduced in this specific version. Previous minor releases as well as patch release 11.0.15.1 should be used instead.
- Java 11.0.15 is not supported due to a defect introduced in this specific version. Previous minor releases as well as patch release 11.0.15.1 should be used instead.
- Amazon Corretto 11.0.16.8.1 is not supported due to a defect introduced in this specific version. Amazon Corretto 17 should be used instead.
- Java 17.0.3 is not supported due to a defect introduced in this specific version. Previous minor releases as well as patch release 17.0.3.1 should be used instead.
- Microsoft OpenJDK builds v11 and v17 are not supported on Windows. These versions contain a known bug that prevents MANTA from working. Other vendors have fixed this.

Bolt port: As of R38, the embedded Neo4j exposes Bolt Protocol for client-server communication. The selected port number (the default value `localhost:7687`) must be unused for MANTA Server to start successfully.

User Access Requirements

A dedicated OS user with limited privileges under which MANTA will run. This is not required, but it is highly recommended to limit MANTA's access to other data and applications for security reasons.

Java Installation Instructions

Every MANTA product requires Java installation. Specifically, Java 64-bit JRE (Java Runtime Environment) version 11 or higher (see the compatibility matrix below) is a prerequisite for MANTA. (See the compatibility matrix.) The current recommended version is Java 11. JRE is NOT part of the MANTA package.It is the sole responsibility of the customer to obtain JRE. The distributions tested by MANTA are OpenJDK and Oracle. Although it is not guaranteed that other distributions of JRE will work with MANTA, distributions based on the two previously mentioned are likely to work fine. For example, IBM JRE is not supported. Each distributor has its own license conditions, and it is up to the customer to fulfill those conditions. MANTA is not in any way responsible for the customer´s compliance with the required conditions.

Installation Instructions

1. Download JRE for your operating system and install it. OpenJDK and Oracle are supported.
2. For OpenJDK on Windows, extract the files in `C:\Programs Files\Java\`.
3. Configure the environment variables—Windows, Unix.
   - `JRE_HOME`—Add the path to where JRE is installed, including the last (back)slash.
     - that is, `JRE_HOME` set to `C:\Program Files\Java\jdk_version\`
   - `PATH`—Add the bin path to where JRE is installed (e.g., `%JRE_HOME%bin`).
     - that is, PATH set to `C:\Program Files\Java\jdk_version\bin`
4. Enable unlimited cryptographic strength for password encryption in Java (see MANTA Flow Password Encryption for more details).
   a. Java 8u151 or newer: Set crypto.policy=unlimited in the file `$JRE_HOME/lib/security/java.security` (there is a template commented out towards the end of the file)
   b. Java 9 - 10: Set `crypto.policy=unlimited` in the file `$JRE_HOME/conf/security/java.security` (there is a template commented out towards the end of the file)
   c. Java 11 or newer: no action needed; unlimited `crypto.policy` is set by default.

MANTA Installation Files

The following files are necessary to proceed with the installation.

- `MantaFlow-x.y-windows-installer-*.exe` (for Windows)
- `MantaFlow-x.y-linux-x64-installer-*.run` (for Linux)
- `license.key` (license file)

On Linux, the specific administrator account called *Root* must be used in order to successfully complete the installation process. On Windows, the user must be a local administrator.

Installation

This page explains how to install MANTA Flow for the very first time. If you want to upgrade an existing MANTA Flow installation, please refer to the MANTA Flow Upgrade Guide (with Installer).

1. Verify that your system meets the MANTA Technical Requirements, which can change when new versions of MANTA are released.
2. Run the installation process.
   a. Windows: run `MantaFlow-x.y-windows-installer-*.exe` using an admin account, and follow the instructions on the screen.
   b. Linux: run `MantaFlow-x.y-linux-x64-installer-*.run` using a root account, and follow the instructions on the screen.
3. Select the preferred Java version for MANTA to use. The installer only offers compatible versions. (As of R36, this is JDK 11.)

4. Make sure you are authorized to accept the license terms MANTA requires to continue with the installation, or ensure that an authorized person in the company has reviewed the MANTA EULA.
5. Select the installation directory for MANTA.
6. Select the components to be installed. (A typical installation includes all components.)
7. Set up a MANTA administrator username and password.
8. Linux: Set the MANTA installation directory ownership to *Non-Root User*, *User with Limited Privileges*.
9. Point the installer to the license key provided by the MANTA Help Desk or the presales team (`license.key` file with an extension/file type showing KEY).
10. Select the port number for the MANTA Server application, and select whether to install it as a service. The service name will be `MantaFlowServer`. It is highly recommended to install the three server processes as a system service so that they start automatically when the machine is rebooted or the user logs out.
11. Select the port number for the MANTA Service application, and select whether to install it as a service. The service name will be `MantaUtilityTool`. It is highly recommended to install the three server processes as a system service so that they start automatically when the machine is rebooted or the user logs out.
12. Select the port number for the MANTA Keycloak application, and select whether to install it as a service. The service name will be `Manta Keycloak`. It is highly recommended to install the three server processes as a system service so that they start automatically when the machine is rebooted or the user logs out.
    a. On the Keycloak configuration page, you have the option to provide the final URL on which the Keycloak server will be accessible. This is useful if there is a DNS, load balancer, or reverse proxy setup. See Networking Setup Examples for more details. When filling in the field, provide the full URL, including the protocol (HTTP or HTTPS) and the port number, if applicable. If the field is left empty, the default value http://localhost:9090 will be used.
13. Complete the installation.
14. The installer creates:
    a. Windows: Shortcuts on the desktop to run MANTA (also present in the mantaflow installation directory)
    b. Linux: Symbolic links to run MANTA in the mantaflow installation directory
15. Verify that MANTA Keycloak is up and running as follows.
    a. If installed as a service, verify the system service `MantaKeycloak`. (The display name on Windows is `MantaKeycloak`.)
    b. Open http://localhost:9090/auth/admin/manta/console in your web browser. Replace *localhost* with the actual machine name or IP address and port *9090* with the actual port number, if changed during the installation.
16. Verify that MANTA Server is up and running as follows.
    a. If installed as a service, verify the system service `MantaFlowServer`. (The display name on Windows is `Apache Tomcat 9.0 MantaFlowServer`.)
    b. Open http://localhost:8080/manta-dataflow-server in your web browser. Replace *localhost* with the actual machine name or IP address and port *8080* with the actual port number, if changed during the installation.
    c. In a Linux shell, it is also possible to check the list of running processes and look for the MANTA Server process; for example, `ps auxw | grep -i "manta.*server"`.
17. Verify that the MANTA Configuration UI is up and running as follows.
    a. If installed as a service, verify the system service `MantaUtilityTool`. (The display name on Windows is Apache Tomcat 9.0 `Manta UtilityTool`.)
    b. Log in to http://localhost:8181/manta-admin-gui/app/#/welcome/login in your web browser. Replace *localhost* with the actual machine name or IP address and port *8181* with the actual port number, if changed during the installation.
    c. In a Linux shell, it is also possible to check the list of running processes and look for the MANTA Service Utility process; for example, `ps auxw | grep -i "manta.*serviceutility"`.
18. Windows: Manually change the MANTA installation directory permissions for a user without system administration privileges that will be used to run MANTA services and processes.
19. If using third-party integrations (Alation, Collibra, Informatica EDC, IBM IGC), please apply the MANTA model as per the instructions for individual third-party connector configurations.
20. If there are any errors, the logs from MANTA installation have the filename mask `installbuilder_installer_*.log` and can be found at:
    - Windows: `<USER_HOME_FOLDER>/AppData/Local/Temp`
    - Linux: `/tmp`
21. Throughout the rest of the documentation, the following placeholders are used.
    a. <MANTA_DIR_HOME> = *installation directory*`/cli`
    b. <MANTA_SERVER_HOME> = *installation directory*`/server`
22. It is strongly advised to change all the default passwords after the installation—specifically, passwords for the truststores used by individual MANTA scanners and integrations.

Upgrade to a Newer Version

This page explains how to upgrade an existing MANTA Flow installation. If no version of MANTA Flow has been installed, please refer to the MANTA Flow Installation Guide (with Installer). Before upgrading, please check the Release Notes for compatibility changes, noted in italics, made to any version newer than the one you are updating from. These changes will need to be reflected in your configuration for MANTA to work after the upgrade. Please allocate extra time and secure any resources needed to make such changes.

Instructions for Versions R36 and Newer

In R36, the Updater component of the installation was removed. The installer now handles the upgrade on its own.

1. Before starting the upgrade:
   a. Verify that your system meets the MANTA Technical Requirements, which may have been updated when new versions of MANTA were released.
      i. Verify that there is enough disk space available—especially for an upgrade from R35 to a newer version, which requires at least as much space as the current Titan database has in total. This is due to the migration from Titan to Neo4J.
   b. You may want to purge old logs to reduce the disk space usage.
   c. Review the Release Notes for every version of MANTA between the one you have now and the one you are upgrading to—especially the notes prefixed with "***Important additional information***", which indicates additional changes that need to be made after the upgrade.
   d. Verify that MANTA Services (Server, Admin UI, Keycloak, and Artemis) are not up and running.
2. Run the installation process.
   a. On Windows, run `MantaFlow-x.y-windows-installer-*.exe` using an admin account, and follow the instructions on the screen.
   b. On Linux, run `MantaFlow-x.y-linux-x64-installer-*.run` using a root account, and follow the instructions on the screen.
3. Select the preferred Java version for MANTA to use. R36 requires at least JDK 11. The installer only offers supported versions. When upgrading from an older version of Java, selecting the desired Java version in the installer is enough. The upgrade JDK used in the platform is handled automatically.
4. Complete the installation.
5. Apply the manual changes described in the Release Notes, which are prefixed with "***Important additional information***".
   a. If using third-party integrations (Collibra, EDC, IGC), please be sure to update the model to the one provided by MANTA with the new release. Do this by following the instructions for individual third-party connector configurations.

Instructions for Versions R35.1 and Older

Upgrade your existing installation of MANTA to a newer version in two steps.

1. Update the service utility, which includes Admin UI with the Configurator and Updater applications.
2. Update the MANTA Flow Server and CLI that host all the scanners and the MANTA metadata repository.

Please see the detailed instructions for each step in the sections below.

Upgrade the MANTA Updater Application

1. Before starting the update:
   a. Verify that there is enough disk space available.
   b. Verify that your system meets the MANTA Technical Requirements, which may have been updated when new versions of MANTA were released.
   c. Review the Release Notes for every version of MANTA between the one you have now and the one you are upgrading to—especially the notes prefixed with "***Important additional information***", which indicates additional changes that need to be made after the upgrade.
   d. Verify that MANTA Server is not up or running as follows.
      i. If installed as a service, verify that the system service `MantaFlowServer` is *Stopped*. (The display name on Windows is `Apache Tomcat 9.0 MantaFlowServer`.)
      ii. You shouldn't be able to reach the MANTA Server site by opening http://localhost:8080/manta-dataflow-server in your web browser. Replace *localhost* with the actual machine name or IP address and port *8080* with the actual port number, if changed during the installation.
      iii. In a Linux shell, it is also possible to check the list of running processes for the MANTA Server process; for example, `ps auxw | grep -i "manta.*server"`.
   e. Verify that the MANTA Configuration UI is not up or running as follows.
      i. If installed as a service, verify that the system service `MantaUtilityTool` is *Stopped*. (The display name on Windows is `Apache Tomcat 9.0 MantaUtilityTool`.)
      ii. You shouldn't be able to reach the MANTA Configurator site by opening http://localhost:8181/manta-admin-gui/app/#/platform/updater in your web browser. Replace *localhost* with the actual machine name or IP address and port *8181* with the actual port number, if changed during the installation.
      iii. In a Linux shell, it is also possible to check the list of running processes for the MANTA Service Utility process; for example, `ps auxw | grep -i "manta.*serviceutility"`.
2. Run the installation process.
   a. On Windows, run `MantaFlow-x.y-windows-installer-*.exe` using an admin account, and follow the instructions on the screen.
   b. On Linux, run `MantaFlow-x.y-linux-x64-installer-*.run` using a root account, and follow the instructions on the screen.
3. Select the preferred Java version for MANTA to use.
4. Decide who should own the files. (Please refer to the installation process for details.)
5. Provide login credentials (username and password) for a MANTA Admin user to upload the update packages. This should be the same login and password you use to log in to http://localhost:8181/manta-admin-gui/app/#.
6. Complete the rest of the process as instructed by the wizard.
7. If there are any errors, the MANTA installation logs have the filename mask `installbuilder_installer_*.log` and can be found at:
   - `<USER_HOME_FOLDER>/AppData/Local/Temp` for Windows

› `/tmp` for Linux

Upgrade MANTA Flow to a Newer Version

1. Optionally, export usage statistics via `<manta-server-url>/usage`.
2. Open http://localhost:8181/manta-admin-gui/app/#/platform/updater. (Replace *localhost* and port *8181* with the actual host name and port number used in your environment.)
3. Use the admin login credentials that were provided during the initial product installation (not the login that is normally used which may be connected to LDAP/AD).
4. To proceed with the update process:
   a. Validate the version column. If it states the target version of the upgrade, the upgrade has been completed. If not, continue with the next step.
   b. If the update packages have been (automatically) uploaded, *Continue* buttons appear next to the Server and CLI applications. Click the *Continue* button for each application and follow the wizard instructions to complete the update process. After completing the process, the button will say *Ready*. No additional action is needed if the version number has been updated to the upgrade target version.
   c. If the update packages have not been automatically uploaded, *Ready* buttons appear next to the Server and CLI applications. If the version number is older than the desired version, click the *Ready* button for each application. On the next screen, upload the respective update packages (they are available in the MANTA installation directory in `/opt/mantaflow/updatepackages/` for Linux and `c:\mantaflow\updatepackages\` for Windows) and follow the wizard instructions to complete the update process.
5. If requested, resolve any configuration conflicts between the existing and new versions of MANTA.
   › A three-way merge screen appears. On the left, it shows the current configuration for the existing version of MANTA. On the right, it shows the default (blank) configuration for the new version you are upgrading to. In the middle is the resulting configuration that will be used after the upgrade. Please merge all the new changes and make sure that all the custom changes/configurations from the existing version are kept so that it works as expected after the upgrade.
   › Please note that only the following file suffixes are considered for the merging process: `.txt`, `.properties`, `.csv`, `.xml`, `.json`, `.ini`, `.config`, `.cfg`, `.sh`, and `.bat`. All the other files will be replaced with newer versions. If you have changed a file with a suffix not listed above, please re-apply the change manually after the update is completed. You can use a backup of a previous installation in `<MANTA_APP>/.backup/<date of update>/<app name>.zip`; however, do not directly copy the files from the backup location because the file structure may have changed in the newer version.
6. Upon successfully completing the update, verify that MANTA Server is up and running as follows.
   a. If installed as a service, verify the system service `MantaFlowServer`. (The display name on Windows is `Apache Tomcat 9.0 MantaFlowServer`.)
   b. Open http://localhost:8080/manta-dataflow-server in your web browser. Replace *localhost* with the actual machine name or IP address and port *8080* with the actual port number, if changed during the installation.
   c. In a Linux shell, it is also possible to check the list of running processes for the MANTA Server process; for example, `ps auxw | grep -i "manta.*server"`.
7. Log out of Admin UI and log back in. The reason why is when MANTA Server was down, a local admin user configured in Admin UI was used for authentication, and this user may not have all the desired user rights (e.g., when using AD/LDAP authentication).
8. Review the Release Notes and adjust your configuration to reflect any compatibility changes, noted in italics, made to any version newer than the one you are updating from.
   a. If using third-party integrations (Collibra, EDC, IGC), please be sure to update the model to the one provided by MANTA with the new release. Do this by following the instructions for individual third-party connector configurations.

Architecture

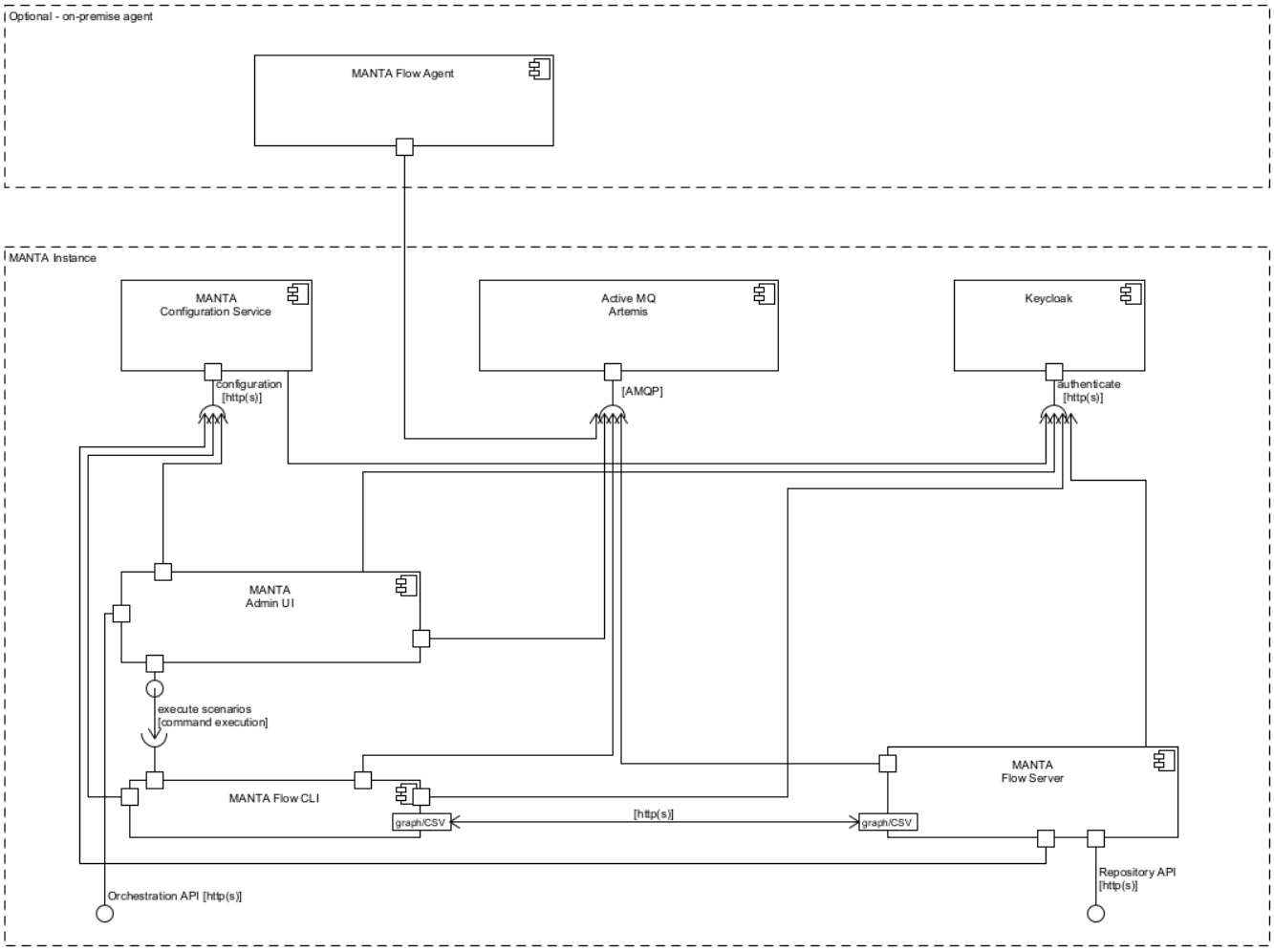
MANTA Flow Components

MANTA Flow is made up of three major components.

› MANTA Flow CLI—Java command line application that extracts all scripts from source databases and repositories, analyzes them, sends all gathered metadata to the MANTA Flow Server, and optionally, processes and uploads the generated export to a target metadata database
› MANTA Flow Server—Java server application that stores all gathered metadata inside its metadata repository, transforms it to a form suitable for import to a target metadata database, and provides it to its visualization or third-party applications via API
› MANTA Admin UI (runs on the MANTA Flow Service Utility application server)—Java server application providing a graphical and programming interface for installation, configuration, updating, and overall maintenance of MANTA Flow (detailed documentation can be found on the page MANTA Admin UI)

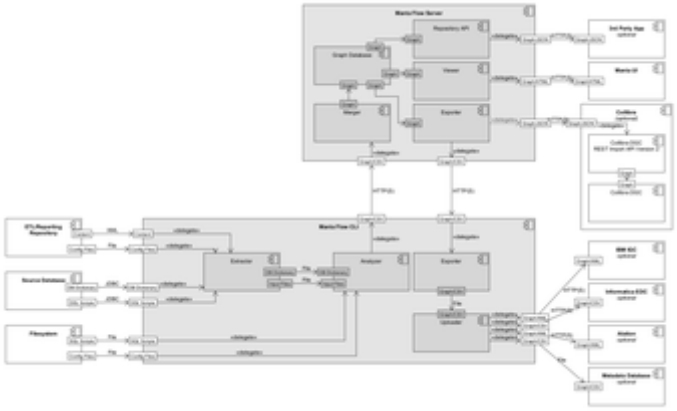
MANTA Flow High-Level Architecture

The following diagram shows the high-level dependencies between the MANTA Flow components mentioned above.



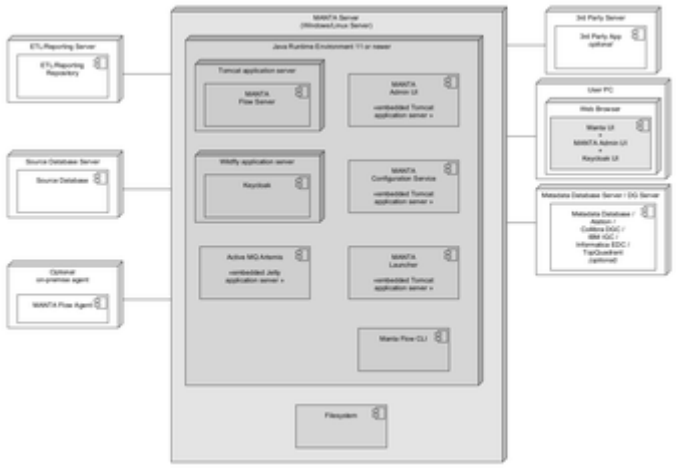
MANTA Flow CLI and MANTA Flow Server Detailed Architecture

The following diagram illustrates MANTA Flow's architecture, components, and interactions with third-party resources.



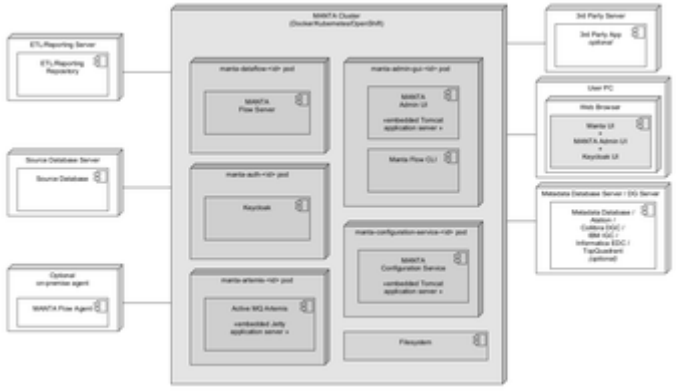
MANTA Flow Deployment Diagram: Installer

All components are deployed to the same physical/virtual server. MANTA UI is accessed through a web-based interface, so there is no need to install anything on the end-user machine or on source systems such as ETL and database servers.



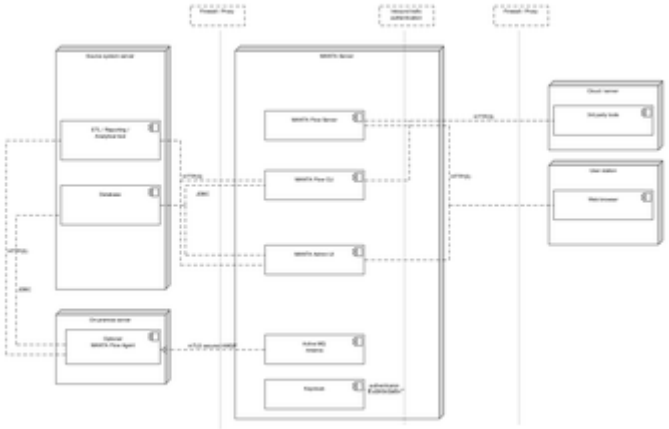
MANTA Flow Deployment Diagram: Containerized

Below is the deployment diagram for containerized MANTA. See MANTA Flow Docker Guides for details.



MANTA Flow Security Architecture

Below is a high-level security architecture diagram. This diagram doesn't consider internal communication between MANTA services and components that are only used internally.



Lineage Scanning Architecture

Overview of the MANTA Flow Scanning Process

What Is a Scan?

MANTA defines a **scan** of a particular technical resource (such as a single database) as the overall process involving the capture, analysis, loading, and merging of lineage metadata into the MANTA Repository. Scans are typically run against a set of connections for a given technology (such as all Oracle databases), but there is a lot of flexibility regarding their execution, which can be invoked by shell script on the

command line, REST API, or by using the graphical user interface. Sites generally determine (based on the needs of their users and the change dynamics of their code and metadata) how frequently to run their scans.

Phases of a Scan

Extract. Metadata is extracted from the host system. Where possible, this is done via direct connection. This ensures that the metadata being retrieved is the most current and most relevant and reflects the truth of how the system is running today. This is usually done by APIs, but it depends on the technology. For most databases, this is done by directly accessing the database catalog. The extract phase first retrieves database dictionary information (primarily tables and columns) and then picks up assets that define lineage. These are typically views and stored procedures for a database but might be external data access steps and transformation modules when applied to an ETL (extraction /transformation/loading) tool. Business intelligence (reporting) solutions deliver lineage details about their queries along with columns in the report and their potential transformations. The extracted metadata is then ready for analysis.

Analyze. Each MANTA **scanner** is developed after extensive research into the selected technology. This research yields a deep understanding of how a particular syntax is derived, what constitutes a source or a target, and how data transformations are defined and stored. Specific dialects for certain languages are explored (such as the subtle differences between SQL-based solutions), and important structural issues are reviewed (a tool or technology's folder, project, and process structure). This allows the scanner, when implemented, to outline the exact flow of data through the selected tool or technology. MANTA strives with every scanner to provide detailed insight for data flows at the individual column or element level, while also capturing exact transformation syntax. During the analyze phase, the dictionary of column information (created earlier during extract) supports validation and proper column usage.

Load-Merge. This phase is where the analyzed metadata is put into the MANTA Repository. New metadata is loaded directly; refreshed metadata is compared to existing assets to detect changes in lineage. These changes can be reviewed later by users who require detailed lineage history. Further, lineage metadata is merged with other lineage metadata that is already in the repository. Names of columns used across multiple technologies are reconciled, allowing connections to be made across different technology types. This ensures the ability to trace the entire data pipeline and achieve end-to-end lineage.

What Is a Revision?

Scans are typically sequenced as part of a full MANTA **run**. A full MANTA run is a coordinated execution of all important scans. The metadata for these scans and their resulting lineage are grouped together in a single **revision**. Each revision represents a snapshot in time of all the coordinated metadata in the corporate systems and technologies used in the enterprise. Users will see the most recent or **current revision** by default, but can easily request lineage or review of an asset at an earlier point in time, supporting any kind of historical lineage research or comparisons.

What Is an Export?

Optionally, some MANTA customers will also take advantage of the export feature. This is the ability to run a specific (licensed by MANTA) technology export or a more generic export. These export capabilities support third-party solutions or other home-grown solutions where MANTA lineage metadata is loaded into other complementary systems. Quite often, these are data catalog solutions, but there are many other examples where MANTA lineage metadata is used to augment other applications. The export phase, once configured, obtains selected MANTA metadata and packages it for shipment to a set of CSV files (in the generic case) or automatically puts it into proprietary API calls (usually a REST API payload in XML or JSON) for a third party and makes the calls on behalf of the MANTA user or site. This loads the lineage metadata into the other application where it can be used for additional purposes.

Details Regarding Individual Phases

Each phase of the scanning process described above is called a **scenario**. Scenarios run sequentially. However, each scenario can be launched for each configuration of the source system in parallel, which is done by **master scenarios**. Therefore, for example, a scenario that extracts DDL scripts from a database runs in parallel for each configured database. The details below describe the activities in each phase as they relate to specific technologies. This is all managed for you by the MANTA Flow Process Manager, but it is described here to provide a deeper understanding of what is occurring as MANTA retrieves and analyzes your lineage metadata.

Extraction Phase

For the extraction phase for Oracle databases, there are two master scenarios.

1. Oracle dictionary mapping master scenario—connects to each configured Oracle database and stores the mapping between these values: global database name, instance name, host name, port, dictionary ID, connection ID, list of schemas
2. Oracle extractor master scenario—connects to each configured Oracle database and extracts the database dictionary and DDL scripts from the configured schemas

For the extraction phase for Teradata databases, there is only one master scenario.

1. Teradata extractor master scenario—connects to each configured Teradata database and extracts the database dictionary and DDL scripts from the configured schemas

For the extraction phase for MS SQL servers, there are two master scenarios.

1. MS SQL dictionary mapping master scenario—connects to each configured MS SQL server and stores the mapping between these values: dictionary ID, subdialect, server name, instance name, host name, port, include and exclude filters
2. MS SQL extractor master scenario—connects to each configured MS SQL server and extracts the database dictionary and DDL scripts from the configured databases and their schemas

For the extraction phase for Hive databases, there is only one master scenario.

1. Hive extractor master scenario—connects to each configured Hive metastore and extracts the database dictionary and DDL scripts from the configured schemas

For the extraction phase for Netezza servers, there are two master scenarios.

1. Netezza dictionary mapping master scenario—connects to each configured Netezza server and stores the mapping between these values: dictionary ID, host name, host port, include and exclude filters
2. Netezza extractor master scenario—connects to each configured Netezza server and extracts the database dictionary and DDL scripts from the configured databases and their schemas

For the extraction phase for DB2 databases, there are two master scenarios.

1. DB2 dictionary mapping master scenario—connects to each configured DB2 database and stores the mapping between these values: dictionary ID, host name, instance name, database name, included and excluded schemas
2. DB2 extractor master scenario—connects to each configured DB2 database and extracts the database dictionary and DDL scripts from the configured schemas

For the extraction phase for PostgreSQL database servers, there are two master scenarios.

1. PostgreSQL dictionary mapping master scenario—connects to each configured PostgreSQL database server and stores the mapping between these values: dictionary ID, subdialect, host name, port, included databases/schemas and excluded databases/schemas
2. PostgreSQL extractor master scenario—connects to each configured PostgreSQL database server and extracts the database dictionary and DDL scripts from the configured schemas

For the extraction phase for SAP HANA database servers, there are two master scenarios.

1. SAP HANA dictionary mapping master scenario—connects to each configured SAP HANA database and stores the mapping between these values: dictionary ID, host name, port, included schemas, and excluded schemas
2. SAP HANA extractor master scenario—connects to each configured SAP HANA database and extracts the database dictionary and DDL scripts from the configured schemas

For the extraction phase for Kafka, there are two master scenarios.

1. Kafka dictionary mapping master scenario—creates the mapping between the dictionary ID and broker URLs for each configured Kafka cluster
2. Kafka extractor master scenario—connects to each configured Kafka Schema Registry server and extracts the schemas

For the extraction phase for the Informatica PowerCenter repository, there are three master scenarios.

1. Informatica PowerCenter service settings extractor master scenario—connects to each configured Informatica PowerCenter domain and extracts the settings of each PowerCenter service; note that source systems are configured on the repository level, so if there are more repositories on the same domain, duplicate extraction occurs which is compensated by an easier configuration
2. Informatica PowerCenter connection extractor master scenario—connects to each configured Informatica PowerCenter repository and extracts all connection definitions
3. Informatica PowerCenter workflow extractor master scenario—connects to each configured Informatica PowerCenter repository and extracts all workflows from all configured folders

Note that the first two steps can also be done manually if necessary.

For the extraction phase for SQL Server Integration Services, there is only one master scenario.

1. SSIS extractor master scenario—connects to each configured SSIS server and extracts the configured packages and/or projects

For the extraction phase for SQL Server Reporting Services, there is only one master scenario.

1. SSRS extractor master scenario—connects to each configured SSRS server and extracts the configured projects

For the extraction phase for Sqoop, there is only one master scenario.

1. Sqoop extractor master scenario—connects to each configured Sqoop repository and extracts the configured scripts

For the extraction phase for Oracle Data Integrator, there is only one master scenario.

1. ODI extractor master scenario—connects to each configured ODI repository and extracts the configured scenarios

For the extraction phase for StreamSets Data Collector, there is only one master scenario.

1. StreamSets extractor master scenario—connects to the configured StreamSets Data Collector and extracts the configured pipelines

For the extraction phase for Snowflake accounts, there are two master scenarios.

1. Snowflake dictionary mapping master scenario—connects to each configured Snowflake account and stores the mapping between these values: dictionary ID, account name, region, connection ID, included databases/schemas and excluded databases/schemas
2. Snowflake extractor master scenario—connects to each configured Snowflake account and extracts the database dictionary and DDL scripts from the configured databases and schemas

For the extraction phase for BigQuery servers, there are two master scenarios.

1. BigQuery dictionary mapping master scenario—connects to each configured BigQuery account and stores the mapping between these values: dictionary ID, service URL, connection ID, included and excluded projects/datasets
2. BigQuery extractor master scenario—connects to each configured BigQuery account and extracts the database dictionary and DDL scripts from the configured projects and datasets

For the extraction phase for Python applications, there is only one master scenario.

1. Python extractor master scenario—extracts the configured Python applications, source code of the provided Python system interpreter and supported Python libraries (if present).

For the extraction phase for Tableau servers, there is only one master scenario.

1. Tableau extractor master scenario—connects to each configured Tableau site and extracts the workbooks and data sources

Analytic Phase

At the beginning of the analytic phase, a new revision scenario is called, which creates a new revision in the internal metadata repository so it is ready to accept new metadata.

For the analytic phase for Oracle databases, there are three master scenarios.

1. Oracle dictionary dataflow master scenario—analyzes metadata from the extracted Oracle database dictionaries and stores it in the internal metadata repository
2. Oracle DDL dataflow master scenario—analyzes metadata from the extracted Oracle DDL scripts and stores it in the internal metadata repository
3. Oracle PL/SQL dataflow master scenario—analyzes metadata from the provided PL/SQL scripts and stores it in the internal metadata repository

For the analytic phase for Teradata databases, there are four master scenarios.

1. Teradata dictionary dataflow master scenario—analyzes metadata from the extracted Teradata database dictionaries and stores it in the internal metadata repository
2. Teradata DDL dataflow master scenario—analyzes metadata from the extracted Teradata DDL scripts and stores it in the internal metadata repository
3. Teradata BTEQ dataflow master scenario—analyzes metadata from the provided BTEQ scripts
4. Teradata Parallel Transporter dataflow master scenario—analyzes metadata from the provided TPT scripts

For the analytic phase for MS SQL servers, there are three master scenarios.

1. MS SQL dictionary dataflow master scenario—analyzes metadata from the extracted MS SQL database dictionaries and stores it in the internal metadata repository
2. MS SQL DDL dataflow master scenario—analyzes metadata from the extracted MS SQL DDL scripts and stores it in the internal metadata repository
3. MS SQL Transact-SQL dataflow master scenario—analyzes metadata from the provided Transact-SQL scripts and stores it in the internal metadata repository

For the analytic phase for Hive databases, there are three master scenarios.

1. Hive dictionary dataflow master scenario—analyzes metadata from the extracted Hive database dictionaries and stores it in the internal metadata repository
2. Hive DDL dataflow master scenario—analyzes metadata from the extracted Hive DDL scripts and stores it in the internal metadata repository
3. Hive HiveQL dataflow master scenario—analyzes metadata from the provided HiveQL scripts

For the analytic phase for Netezza databases, there are three master scenarios.

1. Netezza dictionary dataflow master scenario—analyzes metadata from the extracted Netezza database dictionaries and stores it in the internal metadata repository
2. Netezza DDL dataflow master scenario—analyzes metadata from the extracted Netezza DDL scripts and stores it in the internal metadata repository
3. Netezza PL/SQL dataflow master scenario—analyzes metadata from the provided Netezza PL/SQL scripts

For the analytic phase for DB2 databases, there are three master scenarios.

1. DB2 dictionary dataflow master scenario—analyzes metadata from the extracted DB2 database dictionaries and stores it in the internal metadata repository
2. DB2 DDL dataflow master scenario—analyzes metadata from the extracted DB2 DDL scripts and stores it in the internal metadata repository
3. DB2 PL/SQL dataflow master scenario—analyzes metadata from the provided DB2 PL/SQL scripts and stores it in the internal metadata repository

For the analytic phase for PostgreSQL database servers, there are three master scenarios.

1. PostgreSQL dictionary dataflow master scenario—analyzes metadata from the extracted PostgreSQL database dictionaries and stores it in the internal metadata repository
2. PostgreSQL DDL dataflow master scenario—analyzes metadata from the extracted PostgreSQL DDL scripts and stores it in the internal metadata repository
3. PostgreSQL Plpgsql dataflow master scenario—analyzes metadata from the provided PostgreSQL Plpgsql scripts and stores it in the internal metadata repository

For the analytic phase for SAP HANA databases, there are three master scenarios.

1. SAP HANA dictionary dataflow master scenario—analyzes metadata from the extracted SAP HANA database dictionaries and stores it in the internal metadata repository
2. SAP HANA DDL dataflow master scenario—analyzes metadata from the extracted SAP HANA DDL scripts and stores it in the internal metadata repository
3. SAP HANA SQL dataflow master scenario—analyzes metadata from the provided SAP HANA SQL scripts and stores it in the internal metadata repository

For the analytic phase for Kafka clusters, there is only one master scenario.

1. Kafka dictionary dataflow master scenario—analyzes metadata from the extracted Kafka dictionaries and stores it in the internal metadata repository

For the analytic phase for Informatica PowerCenter repository, there is only one master scenario.

1. Informatica PowerCenter dataflow master scenario—analyzes metadata from the extracted Informatica PowerCenter workflows

For the analytic phase for SSIS server, there is only one master scenario.

1. SSIS server dataflow master scenario—analyzes metadata from the extracted SSIS packages and projects and stores it in the internal metadata repository

For the analytic phase for SSAS server, there is only one master scenario.

1. SSAS server dataflow master scenario—analyzes metadata from the provided SSAS projects and stores it in the internal metadata repository

For the analytic phase for SSRS server, there is only one master scenario.

1. SSRS server dataflow master scenario—analyzes metadata from the extracted SSRS projects and stores it in the internal metadata repository

For the analytic phase for the Pig Latin system, there is only one master scenario.

1. Pig Latin dataflow master scenario—analyzes metadata from the provided Pig Latin systems and stores it in the internal metadata repository

For the analytic phase for Talend projects, there is only one master scenario.

1. Talend dataflow master scenario—analyzes metadata from the provided Talend projects and stores it in the internal metadata repository

For the analytic phase for Azure Data Factory resources, there is only one master scenario.

1. Datafactory dataflow master scenario—analyzes metadata from the provided ADF folders and stores it in the internal metadata repository

For the analytic phase for Sqoop repositories, there are two master scenarios.

1. Sqoop Job dataflow master scenario—analyzes metadata from the extracted Sqoop scripts and stores it in the internal metadata repository
2. Sqoop dataflow master scenario—analyzes metadata from the provided Sqoop scripts and stores it in the internal metadata repository

For the analytic phase for the ODI repository, there is only one master scenario.

1. ODI repository dataflow master scenario—analyzes metadata from the extracted ODI projects and stores it in the internal metadata repository

For the analytic phase for the StreamSets pipeline, there is only one master scenario.

1. StreamSets dataflow master scenario—analyzes metadata from the provided StreamSets pipelines and stores it in the internal metadata repository

For the analytic phase for Fivetran server, there is only one master scenario.

1. Fivetran dataflow master scenario—analyzes metadata from the provided Fivetran folders and stores it in the internal metadata repository

For the analytic phase for ER/Studio models, there is only one master scenario.

1. ER/Studio dataflow master scenario—analyzes metadata from the provided ER/Studio models and stores it in the internal metadata repository

For the analytic phase for Snowflake accounts, there are three master scenarios.

1. Snowflake dictionary dataflow master scenario—analyzes metadata from the extracted Snowflake database dictionaries and stores it in the internal metadata repository
2. Snowflake DDL dataflow master scenario—analyzes metadata from the extracted Snowflake DDL scripts and stores it in the internal metadata repository
3. Snowflake SQL dataflow master scenario—analyzes metadata from the provided Snowflake SQL scripts and stores it in the internal metadata repository

For the analytic phase for BigQuery accounts, there are four master scenarios.

1. BigQuery dictionary dataflow master scenario—analyzes metadata from the extracted BigQuery database dictionaries and stores it in the internal metadata repository
2. BigQuery DDL dataflow master scenario—analyzes metadata from the extracted BigQuery DDL scripts and stores it in the internal metadata repository
3. BigQuery SQL dataflow master scenario—analyzes metadata from the provided BigQuery SQL scripts and stores it in the internal metadata repository
4. BigQuery job dataflow master scenario—analyzes metadata from the provided BigQuery job scripts and stores it in the internal metadata repository

For the analytic phase for Python applications, there is only one master scenario.

1. Python dataflow master scenario—analyzes extracted Python source code and stores the result of the analysis in the internal metadata repository

For the analytic phase for Tableau sites, there is only one master scenario.

1. Tableau dataflow master scenario—analyzes metadata from the provided Tableau workbooks and data sources and stores it in the internal metadata repository

At the end of the analytic phase, six scenarios are called.

1. Import dataflow scenario—imports all Open MANTA metadata to the internal metadata repository
2. Import links dataflow scenario—imports all Open MANTA Direct Links to the internal metadata repository
3. Import metadata dataflow scenario—imports Open MANTA Conceptual Overlay assets with mapping to physical objects to the internal metadata repository
4. Repository post-processing scenario—performs the post-processing of metadata in the internal metadata repository
5. Commit revision scenario—commits the current revision in the metadata repository so the new metadata is accessible to users
6. Prune revision scenario—removes old revisions from the metadata repository

Export Phase

For the export phase, there are several scenarios. Each is used independently, depending on the requirements.

1. Open MANTA basic export scenario—exports all metadata from the internal metadata repository to general CSV files suitable for import to any relational database
2. Open MANTA integration export scenario—exports all metadata from the internal metadata repository to CSV files designed specifically for ingestion to external applications and solutions
3. Alation/Collibra/EDC/IGC—specialized licensed exports that push MANTA lineage metadata to the respective vendor solution repositories

Lineage Scanner Configuration

The entire configuration is done through the UI available at http://localhost:8181/manta-configurator. Replace *localhost* with the actual machine name or IP address and port *8181* with the actual port number if changed during the installation.

The following chapters describe the prerequisites necessary to connect to the source system. The configuration is done in a specific section of MANTA Configurator by creating a new connection or modifying an existing one. Only the chapters that refer to the source systems you choose to connect to are applicable to you. There is also more documentation available with advanced configuration items.

Databases

Use the applicable section of MANTA Configurator to create a new connection to a database or update an existing connection.

You can also define *manual override* to have MANTA recognize the database under a different name or connection string. This is very useful for connecting database lineage with ETL, BI, and reporting tools in cases where MANTA cannot automatically detect the connection. Please follow the instructions in MANTA Configurator if a manual override is needed.

BigQuery

Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- BigQuery service account (an account for apps and virtual machines) with the role ***roles/bigquery.metadataViewer*** (the predefined BigQuery IAM role) for each extracted project
- Connection parameters:
  - Private key from BigQuery service account credentials
  - Client email from BigQuery service account credentials

Supported syntax version:

- As of April 08, 2020

Supported BigQuery job types:

- query

DB2

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

IBM DB2 for Linux, UNIX, and Windows

- IBM DB2 for Linux, UNIX, and Windows: version 11.1 or newer (remote)
- The database must contain the system schema `SYSTOOLS` and table `SYSTOOLS.DB2LOOK_INFO`
  - If the database does not yet contain them, a user with administrative access needs to create them by calling the system procedure `SYSPROC.DB2LK_GENERATE_DDL`. For example:

```
begin
  declare ret int;
  call SYSPROC.DB2LK_GENERATE_DDL ('-e -td ; -z "SYSIBM" -tw "DUAL"',
```

```
ret);
end;
/
```

- User having the following rights:
  - CONNECT ON DATABASE
  - USAGE ON WORKLOAD SYSDEFAULTUSERWORKLOAD
  - EXECUTE ON PROCEDURE SYSPROC.DB2LK_GENERATE_DDL
  - EXECUTE ON PROCEDURE SYSPROC.DB2LK_CLEAN_TABLE
  - USAGE ON SEQUENCE SYSTOOLS.DB2LOOK_TOKEN
  - ALL ON TABLE SYSTOOLS.DB2LOOK_INFO
  - SELECT ON SYSTOOLS.DB2LOOK_INFO_V
  - SELECT on the following system tables and views (unless the database was created in *Restrict* mode, these are granted to every user by default):
    - SYSCAT.ATTRIBUTES
    - SYSCAT.AUDITPOLICIES
    - SYSCAT.COLUMNS
    - SYSCAT.CONDITIONS
    - SYSCAT.CONTEXTS
    - SYSCAT.CONTROLS
    - SYSCAT.DATATYPEDEP
    - SYSCAT.DATATYPES
    - SYSCAT.FUNCTIONS
    - SYSCAT.HIERARCHIES
    - SYSCAT.CHECKS
    - SYSCAT.INDEXES
    - SYSCAT.INDEXEXTENSIONS
    - SYSCAT.INDEXXMLPATTERNS
    - SYSCAT.KEYCOLUSE
    - SYSCAT.MODULEOBJECTS
    - SYSCAT.MODULES
    - SYSCAT.NICKNAMES
    - SYSCAT.PACKAGES
    - SYSCAT.PERIODS
    - SYSCAT.PROCEDURES
    - SYSCAT.REFERENCES
    - SYSCAT.ROLES
    - SYSCAT.ROUTINEDEP
    - SYSCAT.ROUTINEPARMS
    - SYSCAT.ROUTINES
    - SYSCAT.ROWFIELDS
    - SYSCAT.SECURITYLABELCOMPONENTS
    - SYSCAT.SECURITYLABELS
    - SYSCAT.SECURITYPOLICIES
    - SYSCAT.SEQUENCES
    - SYSCAT.SERVEROPTIONS
    - SYSCAT.SERVERS
    - SYSCAT.SCHEMATA
    - SYSCAT.STATEMENTS
    - SYSCAT.TABCONST
    - SYSCAT.TABLES
    - SYSCAT.TABOPTIONS
    - SYSCAT.TRIGGERS
    - SYSCAT.VARIABLEDEP
    - SYSCAT.VARIABLES
    - SYSCAT.VIEWDEP
    - SYSCAT.VIEWS
    - SYSIBM.ROUTINES
    - SYSIBM.SYSDATATYPES
    - SYSIBM.SYSDEPENDENCIES
    - SYSIBM.SYSDUMMY1
    - SYSIBM.SYSFUNCMAPPINGS
    - SYSIBM.SYSFUNCTIONS
    - SYSIBM.SYSINDEXES
    - SYSIBM.SYSINDEXOPTIONS
    - SYSIBM.SYSMODULES
    - SYSIBM.SYSROUTINEOPTIONS
    - SYSIBM.SYSROUTINEPARMS
    - SYSIBM.SYSROUTINES
    - SYSIBM.SYSSERVERS

    - SYSIBM.SYSSCHEMATA
    - SYSIBM.SYSTABLES
    - SYSIBM.SYSTABOPTIONS
    - SYSIBM.SYSTYPEMAPPINGS
    - SYSIBM.SYSUSEROPTIONS
    - SYSIBM.SYSVARIABLES
    - SYSIBM.SYSWRAPPERS
    - SYSIBMADM.ENV_INST_INFO
    - SYSIBMADM.ENV_SYS_INFO
  - Access to an OS (Unix/Windows) as the authentication in DB2 is done through an external security module by a default operating-system-based authentication
- Connection parameters:
  - Name or IP address of the DB2 database
  - Port on which the DB2 database listens for JDBC connections (default is 50000)
  - Name of the database to connect to
  - User name
  - User password
- Database must be accessible via network

Recommended privileges (needed for connection validation, but not for the extraction itself):

SELECT on the following system tables and views (unless the database was created in *Restrict* mode, these are granted to every user by default):

- SYSIBMADM.PRIVILEGES

IBM DB2 for z/OS

Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

- IBM DB2 for z/OS: version 11.0.0 or newer (remote)
- User having the following rights:
  - CONNECT ON DATABASE
  - SELECT on the following system tables and views:
    - SYSIBM.SYSDATATYPES
    - SYSIBM.SYSDEPENDENCIES
    - SYSIBM.SYSDUMMY1
    - SYSIBM.SYSPACKSTMT
    - SYSIBM.SYSROUTINES
    - SYSIBM.SYSPARMS
    - SYSIBM.SYSSCHEMAAUTH
    - SYSIBM.SYSTABLES
    - SYSIBM.SYSSEQUENCES
    - SYSIBM.SYSCOLUMNS
    - SYSIBM.SYSVIEWS
    - SYSIBM.SYSTRIGGERS
    - SYSIBM.SYSVARIABLES
  - SELECT on the following special registers:
    - CURRENT CLIENT_WRKSTNNAME
    - CURRENT MEMBER
    - CURRENT SERVER
- Connection parameters:
  - Name or IP address of the DB2 database
  - Port on which the DB2 database listens for JDBC connections
  - Name of the database to connect to
  - User name
  - User password
- Database must be accessible via network

Hive

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Hive: 3.0 and newer, older versions may or may not work
- Confirm exactly which version of Hive is being used by validating your big data platform vendor and version number or by checking directly by running the query **select version()** against Hive
- Hive JDBC driver provided by your Hadoop platform vendor
- Extract Hive metadata using Hive Server 2 via JDBC
- Kerberos authentication (optional)

Environment with Direct Access to Hive Server 2 via JDBC

- Database user rights:
  - User with SELECT WITH GRANT OPTION rights for all tables in Hive databases that should be extracted
  - MANTA uses these statements to extract metadata: SHOW DATABASES, SHOW CREATE TABLE, SHOW TABLES IN, SHOW FUNCTIONS LIKE, SHOW VIEWS IN
- Connection parameters for the JDBC connection:
  - JDBC driver used
  - Name or IP address of Hive Server 2
  - Port on which Hive Server 2 listens
  - In the case of non-Kerberos authentication
    - User name
    - User password
  - When using Kerberos authentication
    - Principal of Hive Server 2 in the JDBC address
    - Optionally, a keytab and the principal that MANTA should use
  - Hive Server 2 must be accessible via network

Microsoft SQL Server, Azure SQL Database, Azure SQL Data Warehouse, SAP (Sybase) ASE, Parallel Data Warehouse / Analytic Data Platform, and Amazon RDS for SQL Server

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Product support
  - SQL Server: version 2008 or newer (remote)
  - SQL Server: version 2005 (experimental)
  - Analytic Platform System (formerly known as Parallel Data Warehouse)
  - Azure SQL Database
  - Azure SQL Managed Instance
  - Azure Synapse Analytics (formerly known as Azure SQL Data Warehouse)
    - The SQL data warehouse component is supported.
    - Extraction from the "serverless SQL pool" is not fully supported and may produce errors. There may be cases of lineage producing results through TABLE COLUMNS UNKNOWN or an inaccurate view or table column list.
  - Amazon RDS for SQL Server
    - For successful validation and extraction, the following databases have to be added to the "Excluded databases/schemas" list (mssql.excludedDbsSchemas property): model, master, rdsadmin, and msdb.
- User having the following rights:
  - CONNECT SQL—to connect to the server
    - Azure Synapse Analytics specific—only CONNECT SQL is required, no other rights are needed
  - VIEW ANY DATABASE—to extract the list of existing databases
  - ALTER ANY LINKED SERVER—to extract linked server definitions (no write is actually performed, but there's no read-only permission for this metadata)
  - In each extracted database:
    - VIEW DEFINITION (or VIEW ANY DEFINITION on the server level)—to extract object definitions
    - MSSQL = SQL Server (2008 and newer)
      - SELECT on `sys.sql_expression_dependencies`—to extract object dependencies
      - SELECT on the following views in INFORMATION_SCHEMA:
        - COLUMNS
        - TABLES
        - ROUTINES
        - VIEWS
      - SELECT on the following tables in sys:
        - synonyms
        - triggers
        - server_triggers
        - server_sql_modules
        - extended_properties

  - servers
  - all_objects
  - tables
  - index_columns
  - indexes
  - foreign_key_columns
  - sequences
  - table_types
- MSSQL2005 = SQL Server 2005 and older
  - SELECT on `sql_dependencies`—to extract object dependencies
  - SELECT on the following views in INFORMATION_SCHEMA:
    - COLUMNS
    - TABLES
    - ROUTINES
    - VIEWS
  - SELECT on the following tables in sys:
    - synonyms
    - triggers
    - server_triggers
    - server_sql_modules
    - extended_properties
    - servers
    - tables
    - index_columns
    - indexes
    - foreign_key_columns
- PDW = Parallel Data Warehouse
  - SELECT on `sys.sql_expression_dependencies`—to extract object dependencies
  - SELECT on the following tables in INFORMATION_SCHEMA:
    - TABLES
    - VIEWS
    - COLUMNS
    - ROUTINES
  - SELECT on the following tables in sys:
    - synonyms
    - extended_properties
    - tables
    - index_columns
    - indexes
    - foreign_key_columns
- SYBASE = Sybase ASE / SAP ASE
  - SELECT on sysdepends—to extract object dependencies
  - SELECT on:
    - sysobjects
    - sysusers
    - syscolumns
    - systypes
    - syscomments
    - sysdepends
- AZURE_SQL_DB = Azure SQL Database (available as of R34)
  - see MSSQL
- AZURE_MANAGED = Azure SQL Managed Instance (available as of R34)
  - see MSSQL
- AZURE_SYNAPSE = Azure Synapse Analytics (Azure SQL Data Warehouse / SQL Shared pool) (available as of R34)
  - see PDW
- AMAZON_RDS_SQL_SERVER = Amazon RDS for SQL Server (available as of R34)
  - see MSSQL

- Connection parameters:
  - Name or IP address of the SQL Server database
  - Port on which the SQL Server database listens for JDBC connections
  - Mixed mode authentication or Windows authentication
  - User name
  - User password
  - MANTA supports the following authentication types for MS SQL server:
    SQL Server
    The user credentials are configured in the MS SQL server. SQL Server stores both the username and a hash of the password in the master database by using internal authentication methods to verify login attempts. To use this type, fill in the username and password.
    Native (Windows Only)

The identity of the user running MANTA is used to authorize access to the MS SQL server. MS SQL delegates the authentication to the Windows client where the user logged in.
To use native authentication, the property *integratedSecurity=true* has to be set in the JDBC connection string. The Configurator adds this property automatically when the connection is saved.
NTLM (as of MANTA Flow R34)
NTLM authentication allows you to authenticate a user against an active directory. To use NTLM authentication, you can, in addition to the username and password, fill in the domain property which identifies the domain controller.
To use NTLM authentication, the properties *authenticationScheme=NTLM;integratedSecurity=true* have to be set in the JDBC connection string. Configurator adds those properties automatically when the connection is saved.

- Database must be accessible via network
- Linked servers for the OPENQUERY operator
  - In most cases, the linked server used will be extracted from the database.
  - If one of following events is logged, it has to be configured according to `mssql.connections.file` (the requested connection will be part of the log event):
    - ERROR LINKED_SERVER_CONNECTION_NOT_FOUND
    - ERROR MISSING_LINKED_SERVER_CONNECTIONS
    - WARNING NO_MAPPING_FOR_CONNECTION—if the logged linked server data is correct, then this could even be caused by the fact that MANTA has not extracted the targeted linked server
    - WARNING NO_SUBDIALECT_MAPPING_FOR_CONNECTION—if the logged linked server data is correct, then this could even be caused by the fact that MANTA has not extracted the targeted linked server

Netezza

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- IBM PureData System for Analytics: version 7.2 or newer (remote)
- User having the following rights for each extracted database:
  - LIST—for connecting to the database
  - SELECT on the following system tables / views (granted to every user by default, unless explicitly revoked):
    - _t_environ
    - _v_database
    - _v_schema
    - _v_table
    - _v_extobject
    - _v_view
    - _v_relation_column
    - _v_relation_keydata
    - _v_procedure
    - _v_function
    - _v_aggregate
    - _v_sequence
    - _v_synonym
  - In each extracted schema:
    - LIST—to access the schema
    - SELECT on the following management table (must be explicitly granted):
      - _vt_sequence
    - LIST on all relevant object types:
      - `GRANT LIST ON <db>.<schema>.SEQUENCE TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.SYNONYM TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.TABLE TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.EXTERNAL TABLE TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.FUNCTION TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.AGGREGATE TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.PROCEDURE TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.VIEW TO <manta user>;`
      - `GRANT LIST ON <db>.<schema>.MATERIALIZED VIEW TO <manta user>;`
- Connection parameters:
  - Name or IP address of the Netezza database
  - Port on which the Netezza database listens for JDBC connections (default is 5480)
  - Initial database (can be any of the extracted databases)
  - User name
  - User password
- Database must be accessible via network

Oracle

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Oracle: version 11, 12, or 18 (remote)
- User having the following rights:
  - CONNECT and SELECT_CATALOG_ROLE privileges
- Connection parameters:
  - Name or IP address of the Oracle database
  - Port on which the Oracle database listens
  - User name
  - User password
  - SID or service name
- Database must be accessible via network
- MANTA Flow supports scanning SQL and PL/SQL statements/scripts—but not SQL*Plus. Please reference How to Pre-Process Oracle and SQL*Plus Scripts.

PostgreSQL, Greenplum, Yellowbrick, Redshift, Amazon RDS, or Amazon Aurora for PostgreSQL

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- PostgreSQL: version 8.2 or newer (remote)
- Greenplum: version 4.3 or newer (remote)
- User having the following rights for each extracted database:
  - CONNECT—to connect to the database
  - SELECT on the following tables and views in `pg_catalog` (usually granted implicitly through the *Public* role):
    - pg_database
    - pg_namespace
    - pg_class
    - pg_attribute
    - pg_type
    - pg_proc
    - pg_language
    - pg_description
    - pg_constraint (PostgreSQL only)
    - pg_shdescription (PostgreSQL and Greenplum only)
    - svv_external_schemas (Redshift only)
    - svv_external_tables (Redshift only)
    - svv_external_columns (Redshift only)
  - USAGE on all analyzed external schemas (Redshift only)
- Connection parameters:
  - Name or IP address of the database
  - Port on which the database listens for JDBC connections
  - User name
  - User password or authentication token if an Amazon RDS or Amazon Aurora for PostgreSQL “Password and IAM database authentication” authentication type is used
- Database must be accessible via network
- Amazon RDS or Amazon Aurora for PostgreSQL database authentication type:
  - Password authentication
  - Password and IAM database authentication
- AWS JDBC Driver for PostgreSQL
  - The MANTA Flow distribution does not contain an AWS JDBC driver that works with the Amazon Aurora for PostgreSQL database. Please download the most recent compatible version of the JDBC driver from the AWS JDBC Driver for PostgreSQL website (https://awslabs.github.io/aws-postgresql-jdbc/), and place the new driver file `aws-postgresql-jdbc-<version>.jar` in the `cli/scenarios/manta-dataflow-cli/lib-ext` directory.

Snowflake

Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

Please note that for the procedures defined in Javascript, data flow via those procedures is not supported.

- Permission to access all addresses as described in https://docs.snowflake.com/en/user-guide/hostname-whitelist.html
- Snowflake account (the root account representing a server provided by Snowflake)
- User with one of the following two roles:
  - Role with the following privileges for each extracted database (`INFORMATION_SCHEMA` mode):
    - `USAGE` on extracted database
    - `USAGE` on each extracted schema inside the extracted database
    - At least one privilege (it does not matter which one; for example, `SELECT`) on each extracted Snowflake object (for example, TABLE, VIEW)
    - `MONITOR EXECUTION` for extraction of tasks from TASK_HISTORY (as of 3.38)
  - Role with the following privilege (`ACCOUNT_USAGE` mode):
    - `IMPORTED` privilege on the database `Snowflake` (statement: `grant imported privileges on database snowflake to <role>`)
- Connection parameters:
  - Snowflake account containing the following information:
    - Account name
    - Region (optional; check your cloud platform at https://docs.snowflake.com/en/user-guide/jdbc-configure.html#connection-parameters and provide the region if a corresponding “Full Account Name” format contains it)
    - Cloud platform (optional; check your cloud platform at https://docs.snowflake.com/en/user-guide/jdbc-configure.html#connection-parameters and provide the cloud platform if a corresponding “Full Account Name” format contains it)
  - User name
  - Password
    - Not needed if key pair authentication is used
  - Private key file (as of 3.32)
    - Path to a file storing a key
    - Only needed if a key pair authentication is used
  - Private key file password (as of 3.32)
    - Password to the file storing the key
    - Only needed if a key pair authentication is used

Supported Features

- Snowflake SQL (the majority of the language constructs with lineage impact)
- SQL stored procedures (using Snowflake scripting code)

Examples of Unsupported Features

- Snowpark (Java, Scala, Python)
- Javascript Snowflake procedures
- UDFs (User Defined Functions) in Java, Python, Javascript
- Secure UDFs
- Lineage through dynamically executed code through EXECUTE IMMEDIATE
- Recently introduced or changed features and some features that are rarely used

Known Issues

- Snowflake JDBC driver is not compatible with Java 15 and newer. Java 11 should be used until Snowflake provides a JDBC driver compatible with Java 17.
- Tasks are extracted from TASK_HISTORY in both extraction modes, as Snowflake currently (as of 3.38) does not provide appropriate views to extract defined tasks.
  - In `INFORMATION_SCHEMA` mode: TASK_HISTORY returns task activity within the last seven days or the next scheduled execution within the next eight days. It is limited to 10,000 records, showing executions with the most recent timestamp.
  - In `ACCOUNT_USAGE` mode: TASK_HISTORY returns the history of task usage within the last 365 days (one year).

Teradata

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Teradata: version 12, 13, 14, 15, 16, 17 including Vantage (remote)
- User having the following rights:
  - Read-only access to the DBC database (tables DBase, TVM, TVFields, All_RI_ChildrenV, indexes)
  - Permission to call SHOW VIEW, SHOW MACRO, and SHOW PROCEDURE statements
  - Creation of volatile tables

- Connection parameters:
  - Name or IP address of the Teradata server
  - Port on which the Teradata server listens
  - User name
  - User password
  - Database where the volatile tables are created (usually the same as the user name)
- Server must be accessible via network
- Teradata JDBC driver
  The MANTA Flow client distribution might not contain Teradata JDBC drivers that work with the version of the Teradata database you have, or it might not contain the most recent version of those drivers. Check if the files `terajdbc4-<version>.jar` and `tdgssconfig-<version>.jar` (note that the latter JAR file is no longer needed when downloading a 16.20.00.12 or newer JDBC driver package) are present in the directory `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/lib-ext` (`<MANTA_DIR_HOME>/scenarios/manta-dataflow-xxx-cli/lib` prior to 3.32) in the most recent compatible version. If not, please download the most recent compatible version of the JDBC drivers from the Teradata website (https://downloads.teradata.com/download/connectivity/jdbc-driver), remove the old drivers from the `lib` directory, and place the new drivers there in the form of two JAR files.
- BTEQ scripts
  - BTEQ scripts available on the filesystem in the form of files
  - BTEQ scripts need to be pure BTEQ—no Linux shell, PowerShell, or batch file wrappers are supported
  - Parameter mapping if placeholders are used in BTEQ scripts; a mapping to actual values in the format `placeholderName=actualValue` must be available in the form of a file on the filesystem
- TPT scripts
  - TPT scripts available on the filesystem in the form of files
  - TPT scripts need to be pure TPT—no Linux shell, PowerShell, or batch file wrappers are supported

SAP HANA

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

This is an early implementation of the SAP HANA scanner with only a subset of the DDL and DML statements.

Statements that are supported include: CREATE TABLE, CREATE VIEW (parametrized and calculation views not supported), CREATE FUNCTION/PROCEDURE, CREATE TRIGGER, DO BEGIN ... END, TRUNCATE TABLE, SELECT, INSERT, DELETE, UPDATE, UPSERT, CALL, IF/WHILE/FOR, MERGE INTO, ALTER TABLE, CREATE TYPE, RENAME COLUMN.

Statements that are currently unsupported include: EXPORT INTO, CREATE SEQUENCE, CREATE DATABASE, CREATE SCHEMA, GRAPH WORKSPACES, CREATE INDEX

- SAP HANA on premises 2.0 SPS 05 or newer or SAP HANA Cloud
  - The differences between the on-premises and cloud versions are described in https://help.sap.com/viewer/3c53bc7b58934a9795b6dd8c7e28cf05/hanacloud/en-US/a55dd4eb896c4198828f179f4a12feab.html.
- SAP HANA JDBC driver
  The MANTA Flow client distribution does not contain SAP HANA JDBC driver. Please use the JDBC driver available with the SAP HANA client (https://help.sap.com/viewer/product/SAP_HANA_CLIENT/2.9/en-US) and copy the driver JAR file to `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/lib-ext`.
- User having the following rights:
  - PUBLIC role (usually assigned implicitly)
  - **or** CATALOG READ system privilege with SELECT granted on the following system views
    - SYS.TABLES
    - SYS.TABLE_COLUMNS_ODBC
    - SYS.VIRTUAL_TABLES
    - SYS.VIEWS
    - SYS.VIEW_COLUMNS
    - SYS.VIEW_PARAMETERS
    - SYS.FUNCTIONS
    - SYS.FUNCTION_PARAMETERS
    - SYS.FUNCTION_PARAMETER_COLUMNS
    - SYS.PROCEDURES
    - SYS.PROCEDURE_PARAMETERS
    - SYS.PROCEDURE_PARAMETER_COLUMNS
    - SYS.LIBRARIES
    - SYS.OBJECT_DEPENDENCIES
    - SYS.REMOTE_SOURCES
    - SYS.SCHEMAS
    - SYS.SEQUENCES
    - SYS.SYNONYMS
    - SYS.TRIGGERS

- Connection parameters:
  - Name or IP address of the database
  - Port on which the database listens for JDBC connections
  - User name
  - User password
- Database must be accessible via network

ETL Tools

IBM DataStage

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee the success of the connection or integration since we have no control, liability, or responsibility for third-party products or services, including their performance.

- IBM InfoSphere DataStage: version 11.7 (other versions may work but haven't been tested)
- Currently, only parallel jobs are supported and can be imported to MANTA.
- DataStage parallel job metadata must be exported using the "xml" option in order to be processed by MANTA.
- The appropriate `*.xml` DataStage export files must be placed in the folder specified by the configurable `datastage.input.dir` property, which is by default set to <MANTAFLOW_INSTALLATION_DIRECTORY>/cli/input/datastage/<CONNECTION_NAME>. A best practice is to create a separate connection for each DataStage project. So, make the "DataStage server name" the name of the DataStage project. (The wording for the property can be misleading; it should be considered a project name.)
  - For successful analysis of export files, the minimum memory allocation for MANTA CLI should be at least 20x the size of the largest export file. You can change the memory size in `mantaflow/cli/platform/bin/mantar.sh` (if you run CLI jobs using shell script / batch files) by adjusting the parameter `-Xmx` in the line
    `set MEMORY_OPTS=-Xms128m -Xmx3072m -Xss4m`
    More information can be found on the page dedicated to setting memory allocation for MANTA resources.
- If you want to use project/environment level parameters in your job and then let MANTA analyze it, provide MANTA with a DSParams file, which you can get from the project directory or from the administrator client in the environment variables windows using the *Export to File...* function as shown above. After that, rename this file DSParams, if necessary, and put it at the path `${manta.dir.input}/datastage/${datastage.extractor.server}/DSParams` (create folders on the path if needed).
- If you are using parameter sets in your job and then let MANTA analyze it, set the name of the value file that you want to use for the analysis; otherwise the default value file will be used. Set the name of the value file in the `datastage.value.files` parameter in the DataStage configuration.
- If you want to use Operational Metadata (OMD files) collected during job runs to resolve any type of parameter in your jobs, put OMD files in the folder defined in the `datastage.omd.files.directory` property.
- As of MANTA Flow 1.34, if you want to add new parameters to your project or override old ones, create a `datastageParameterOverride.txt` file in the location defined by the `datastage.parameter.override.file` property and fill it out according to the format defined below in the section describing parameter override file.
  - As of MANTA Flow 1.35, after each analysis, when any unresolved parameters are found, a "helper" file `datastageParameterOverride.txt` is generated in the location defined by the `datastage.parameter.override.helper.file` property. The hint that helper file gives can also be found in the log file relevant for the current DataStage dataflow analysis. You can replace the original file with this "helper" file and complete the missing parameter values. Detailed instructions can be found in the section describing parameter override file.
- For unsupported stages, which use the wise method (find out more about this method below), you can link input and output columns manually instead of using the wise method as defined below. For manual mapping, create a CSV file defined by the `datastage.manual.mapping.file` property and fill it out according to the format defined below in the paragraph about manual mapping files.
- If you are loading SQL queries from files in your database connectors (Oracle, DB2, etc.) and you have MANTA installed on a server other than MANTA Server, MANTA won't be able to find the SQL files, so you need to provide MANTA with the SQL files for correct analysis. The SQL files should be placed in the directory defined by the `datastage.sql.files.directory` property.

Supported Features

Here is a list of supported parallel stages. (Unlisted stages may be supported somehow, but no guarantees are provided. Also, if MANTA does not provide support for a third-party technology listed below, such as IWay or Classic Federation, then the flow analysis will end inside DataStage.)

> **Note**
> This is a list of DataStage features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

Processin	Supported

File stage	Supported
Amazon S3	✅
Complex Flat File	✅

Database stage	Supported
Classi	⚠

g stage	
Aggregator	✓
Bloomfilter	✓
Change Apply	✓
Change Capture	✓
Checksum	✓
Compare	✓
Compress	✓
Copy	✓
Data Masking	✓
Decode	✓
Difference	✓
Encode	✓
Expand	✓
External Filter	⚠ An external filter can be anything, so we provide a wise column connection of input and output columns. Find out more about the wise approach below.
Filter	✓
FTP Enterprise	✓
FTP Plu	✓

Data Set	✓
External Source	✓
External Target	✓
File Connector	✓
File Set	✓
Lookup File Set	✓
Sequential File	✓
Unstructured Data	✓
zOS File	✓

Container stage	Supported
Local Container	✓
Shared Container	✓

c Federation	The flow will end in DataStage because MANTA doesn't support the Classic Federation database.
Cognos TM1	⚠ The flow will end in DataStage because MANTA doesn't support the Cognos TM1 tool.
DB2	✓
Distributed Transaction	✓
DRS	✓
Greenplum	✓
HBase	⚠ The flow will end in DataStage because MANTA doesn't support the HBase database.
Hive	✓
Informix Bulk Load	⚠ The flow will end in DataStage because MANTA doesn't support the Informix tool.
Informix CLI	⚠ The flow will end in DataStage because MANTA doesn't support the Informix tool.
Informix Enterprise	⚠ The flow will end in DataStage because MANTA doesn't support the Informix tool.
IWay	⚠ The flow will end in DataStage because MANTA doesn't support the IWay database.
JDBC	✓
MS SQL	✓
Netezza	✓
ODBC	✓
Oracle	✓
Stored Procedure	⚠ MANTA doesn't support stored procedures for Sybase and Teradata.

gin	
Funnel	✅
Generic	⚠️ A generic operator can be any orchestrate operator, so we provide a wise column connection of input and output columns. Find out more about the wise approach below.
Join	✅
Lookup	✅
Merge	✅
Modify	✅
Pivot	✅
Pivot Enterprise	✅
Remove Duplicates	✅
Slowly Changing Dimension	✅
Sort	✅
Surrogate Key Generator	✅
Switch	✅
Transformer	⚠️ DS Routines and functions in expressions are resolved as direct flows.
Wave	✅
Sybase Bulk Load	✅
Sybase Enterprise	✅
Sybase OC	✅
Teradata	✅

Generator	

Wise Column Connection

Wise column connection is the approach where columns with the same names for the input and output are connected. Otherwise, if the input column doesn't have a corresponding output column, it will be connected to all output columns. And if the output column doesn't have a corresponding input column, it will be connected to all input columns.

As of MANTA Flow 1.34, you can change the behavior of this approach by setting the common property `datastage.unsupported.stages.connect.all` to false. This will cause input columns that don't have corresponding output columns to be connected to all output columns, except those that have the same name for the input and output. The behavior of the output columns is similar.

Examples of Unsupported Features

Note
Here are some examples of DataStage features that MANTA doesn't support yet. Please note that this list is probably incomplete, as there might be DataStage features that we don't know of. Most of the features listed will be supported in future releases.

- Common data sorting—https://www.ibm.com/support/knowledgecenter/en/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/sortingdata.html
- Runtime column propagation—https://www.ibm.com/support/knowledgecenter/en/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Runtime_Column_Propagation.html
- `.dsx` export format

Informatica PowerCenter

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Informatica PowerCenter: version 9.x or 10.x (remote)
- User having the following rights:
  - Read privileges for all folders and workflows in the repository
  - Read privileges for connection settings
  - Read privileges for integration service settings
- Connection parameters:
  - Domain name
  - Host of the PowerCenter server
  - Port of the PowerCenter server
  - User name
  - User password
  - Repository name
- PowerCenter must be accessible via network
- PowerCenter client tools must be properly installed and configured on the same machine as MANTA
  - For Windows implementations, this includes a complete Informatica PowerCenter and Developer Client installation
  - For Linux implementations, this includes either a full install of PowerCenter and Developer Client or the PowerCenter Command Line Utilities package
  - Note that MANTA Flow only uses the following two command line tools (the full installs noted above are required to properly utilize these components):
    - Pmrep tool for workflow and connection setting extraction
    - Infacmd tool for integration service setting extraction
- Parameter and indirect files must be available on the MANTA machine local filesystem in the form of files

PowerCenter Scanner Overview

The PowerCenter scanner extracts XML specifications for workflows and mappings from the PowerCenter repository as well as connection definitions and integration service parameters. It analyzes workflows and all the objects they depend on and produces data lineage. If the PowerCenter scanner and relevant database scanners are configured properly, MANTA Viewer can display lineage starting at the source databases/files/applications and flowing through all the transformations to the target databases/files/applications.

Supported Features

Note

This is a list of PowerCenter features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

- Workflows, worklets—including workflow variables, integration service parameters, and parameter files (also multiple parameter files for one workflow)
- Sessions, mappings, mapplets—including mapping parameters and parameter files
- Source definitions
  - File Reader—direct, indirect, command only via customization
  - Relational Reader
  - XML Reader
  - Teradata Parallel Transporter Reader
  - JMS Reader—as an application
  - JDBC Reader
  - Netezza Bulk Reader
  - PowerExchange Reader for sequential files—as an application
  - PowerExchange Batch Reader for VSAM files—as an application
  - Salesforce Reader—as an application
  - PowerExchange CDC Reader for ORACLE—does not support SQL query override
- Target definitions
  - File Writer—supports different merge types and external loaders, doesn't support dynamic filename
  - Relational Writer—supports pre/post SQL attributes
  - XML Writer
  - Teradata Parallel Transporter Writer
  - JMS Writer
  - JDBC Writer
  - Vertica Bulk Writer
  - Netezza Bulk Writer
  - PowerExchange Writer for sequential files—as an application
  - Salesforce Writer—as an application
- Standard transformations
  - Lookup procedure—relational (supports lookup SQL override), file (direct, indirect)
  - Source qualifier—supports SQL query and pre/post SQL attributes
  - Application source qualifier
  - App multi-group source qualifier
  - XML source qualifier
  - Stored procedure—no stored procedure parsing
  - External procedure—no stored procedure parsing
  - Expression
  - Joiner
  - Aggregator
  - Router
  - Sorter
  - Filter
  - Normalizer—does not support VSAM Normalizer
  - Rank
  - Sequence
  - Transaction control
  - Update strategy
- Custom transformations
  - Union
  - HTTP transformation—partial support
  - SQL transform—does not support the SQL query attribute
  - XML generator
  - XML parser
  - Consolidate—only selected versions
  - ExternalChecker—only selected versions
  - Manual definition for data lineage supported

Examples of Unsupported Features

Note
This is a list of PowerCenter features that MANTA doesn't support yet. Please note that this list is probably not complete, as there might be PowerCenter features that we don't know of. Most of the features listed below will be supported in future releases.

- Source/Target definitions
  - WebServices Consumer Reader/Writer
  - WebServices Provider Reader/Writer
- Transformations

- Java
- Salesforce
- Unstructured data
- Data masking

Microsoft SSIS

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- SQL Server Integration Services: version 2012 or newer
- Server must be accessible via TCP
- If using the project deployment model:
  - User with SQL Server authentication and either the *sysadmin* role, *ssis_admin* role, or all the following privileges
    - SELECT on tables/views
      - `SSISDB.catalog.projects`
      - `SSISDB.catalog.folders`
      - `SSISDB.catalog.packages`
      - `SSISDB.catalog.environment_references`
    - EXECUTE on `SSISDB.internal.get_project_internal`
    - EXECUTE on `SSISDB.catalog.get_parameter_values`
    - READ on every extracted project
    - READ on every extracted project's folder
- If using the package deployment model with packages deployed in SQL Server:
  - User with SQL Server authentication and either the `db_ssisoperator` role or all the following privileges
    - SELECT on `msdb.dbo.sysssispackages`
    - SELECT on `msdb.dbo.sysssispackagefolders`
  - Access to any additional configuration files stored on the server's file system
- Connection parameters:
  - Name or IP address of the SQL Server database where SSIS is deployed
  - Port on which the SQL Server database listens for JDBC connections
  - User name
  - Password
  - MANTA supports the following authentication types for MS SQL server:
    SQL Server
    The user credentials are configured in the MS SQL server. SQL Server stores both the username and a hash of the password in the master database using internal authentication methods to verify login attempts. To use this type, fill in the username and password.
    Native (Windows Only)
    The identity of the user running MANTA is used to authorize access to the MS SQL server. MS SQL delegates the authentication to the Windows client where the user logged in.
    To use native authentication, the property `integratedSecurity=true` has to be set in the JDBC connection string. The Configurator adds this property automatically when the connection is saved.
    NTLM (as of MANTA Flow R34)
    NTLM authentication allows you to authenticate a user against an active directory. To use NTLM authentication, you can, in addition to the username and password, fill in the domain property that identifies the domain controller.
    To use NTLM authentication, the properties `authenticationScheme=NTLM;integratedSecurity=true` have to be set in the JDBC connection string. Configurator adds those properties automatically when the connection is saved.

Examples of Unsupported Features

> **Note**
> These are some examples of SSIS features that MANTA doesn't support yet. Please note that this list is not complete. We plan to support most of the listed features in future releases.

- Third-party components
  - KingswaySoft components (CRM source and destination)
    - `KingswaySoft.IntegrationToolkit.DynamicsCrm.CrmSourceComponent`
    - `KingswaySoft.IntegrationToolkit.DynamicsCrm.CrmDestinationComponent`
  - Task Factory components
    - `PW.TaskFactory.DimensionMergeSCD`
    - `PW.TaskFactory.UpsertDestination`

Oracle Data Integrator

The following are the prerequisites necessary to connect MANTA to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Oracle Data Integrator: 12 (remote)
- ODI repository must be accessible via TCP
- When the ODI repository is hosted in database technology that MANTA does not bundle the JDBC driver for (see Third-Party JDBC Drivers and any technology not mentioned there), then the JDBC driver must be placed in the `<MANTA_DIR_HOME>/scenarios/manta-dataflow-xxx-cli/lib-ext` directory (`<MANTA_DIR_HOME>/scenarios/manta-dataflow-xxx-cli/lib` in versions prior to R32)
- User having rights to read ODI repository tables (including all the `SNP_*` tables)
- Connection parameters:
  - JDBC connection string to the ODI repository
  - JDBC driver name
  - User name
  - Password
  - Encoding of ODI projects

Talend

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee the success of the connection or integration since we have no control, liability, or responsibility for third-party products or services, including their performance.

- Talend Data Integration: version 6.5 or newer
- Talend job exports in XML format (`*.item`) or Talend job exports in archive file format (`*.zip`) containing job exports in XML format

MANTA supports analysis of both Talend export formats (zip and item), but there can be differences in the results.

- Talend jobs provided as individual `*.item` files are analyzed independently, and therefore, no direct relationship between job or joblet components in different jobs is created.
- When a zip archive file is analyzed, the whole zip archive file with all the jobs and joblets inside it is analyzed as one task, and therefore, the relationships between jobs will be analyzed as well. In this case, a relationship is, for example, when one job calls another job through a tRunJob component with the results of the called job propagated to the job that is calling. In the zip archive format, data lineage is created between the result of the called job and tRunJob.

Please note that the content and structure of the Talend archive file format export must comply with the same directory and file structure as generated by Talend Studio.

Exporting jobs

Jobs have to be exported and provided manually for the dataflow analysis, see the process.

By default, assuming that `<proj_name>` is the name of the project, the exported files related to jobs are located in `/../<proj_name>/process`. Inside the folder `process`, jobs can be further organized into subfolders. All files related to one job are always located in the same (sub)directory and have the same filename (differing only in extension) — this is mandatory, as otherwise, Talend Studio is unable to import the job back into the workspace.

The whole folder `<proj_name>` can also be exported as a `.zip` archive.

Building jobs

Talend Studio allows to build jobs as standalone units containing scripts for their execution or scheduling (.bat file for windows), see the process. The structure is very similar to one of the exported jobs. By default, the built job is zipped into <job_name>.zip. Among others, the archive contains the folder `<proj_name>` where job definitions are stored as described in the previous section. However, it is important to note that when setting the build properties, the user can choose not to include the .item files, at which point the job definitions are not stored in the build – that is not desirable.

If the built job references (depends on) other jobs within the same project (i.e. it calls some child jobs as a part of its ETL operations), then all necessary definitions of those referenced are automatically also included in the build.

Pig Latin

Note
Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary to connect MANTA to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including their performance.

- Pig Latin: version 0.x
- Pig Latin scripts and macros

Sqoop

> **Note**
> Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary to connect MANTA to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Sqoop: 1.4.x
- Sqoop repository must be accessible via TCP
- Configured JDBC driver must be present in the `<MANTA_DIR_HOME>/scenarios/manta-dataflow-xxx-cli/lib-ext` directory (`<MANTA_DIR_HOME>/scenarios/manta-dataflow-xxx-cli/lib` in versions prior to 3.32)
- User having rights to read Sqoop repository tables
- Connection parameters:
  - JDBC connection string to the Sqoop repository
  - JDBC driver name
  - User name
  - Password
  - Encoding of Sqoop scripts

StreamSets

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee the success of the connection or integration since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- StreamSets Data Collector (SDC): version 3.10.1 (other versions greater than 3.0.0 should work but haven't been tested)
- If Control Hub is **not** enabled
  - Connection parameters to StreamSets Data Collector
    - Address
    - Port
    - Scheme (HTTPS is supported as of MANTA Flow 1.32)
    - User name
    - Password
  - StreamSets Data Collector must be accessible via network
- If Control Hub Cloud is enabled (supported as of MANTA Flow 1.33)
  - Credentials to the Control Hub account
    - User ID
    - Password
- If Control Hub On-Premises is enabled (supported as of MANTA Flow 1.35)
  - Connection parameters to StreamSets Control Hub
    - Address
    - Port
    - Scheme
  - Credentials to the Control Hub account
    - User ID
    - Password
- StreamSets objects can also be exported manually. In such cases, the appropriate `*.json` StreamSets export files must be placed in the MANTA CLI temp folder in the subfolder with the name matching the model system. A best practice is to create a subdirectory with the same name as the StreamSets project. `${manta.dir.temp}/streamsets/${streamsets.extractor.server}`. Inside this directory, two subdirectories have to be created by the user:
  - `/pipelines`—this directory should contain all exported pipelines
  - `/jobs`—this directory should contain all exported jobs
- If you use the runtime values in the pipelines and you want to include these values in MANTA's analysis, supply the properties file and other files with the runtime values to `${manta.dir.input}/streamsets/${streamsets.extractor.server}`.
  - The runtime values are called with the functions `${runtime:conf(<property name>)}` and `${runtime:loadResource(<file name>, <restricted: true | false>)}`.
  - The path to the runtime properties is `$SDC_CONF/sdc.properties` (MANTA expects `*.properties`) and to the SDC runtime resources is `$SDC_RESOURCES`. For more information about SDC Directories go to *Administration SDC Directories* in SDC.

Supported Features

This is a list of the StreamSets features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

Here is a list of supported stages. (Default visualization without any data flow analysis is provided for unlisted stages.)

Sources: Stage name	Supported
Directory	✅
Hadoop FS Standalone	✅
JDBC Query Consumer	✅
Kafka Multitopic Consumer	⚠️ MANTA doesn't extract metadata from Kafka, so the analysis might be incomplete.
Salesforce	⚠️ MANTA doesn't support Salesforce Object Query Language (SOQL). Default SQL analysis is provided.
Oracle CDC Client	✅
PosgreSQL CDC Client	✅
Google BigQuery (as of MANTA Flow 1.32)	✅

Destinations: Stage name	Supported
Hadoop FS	✅
Hive Metastore	✅
HTTP Client	✅
Kafka Producer	⚠️ MANTA doesn't extract metadata from Kafka, so the analysis might be incomplete.
Local FS	✅
Trash	✅
JDBC Producer	✅
Google BigQuery (as of MANTA Flow 1.32)	✅
Snowflake (as of MANTA Flow 1.35)	✅

Processors: Stage name	Supported
Expression Evaluator	✅
Field Hasher	✅
Field Masker	✅
Field Order	✅
Field Remover	✅
Field Renamer	✅

Field Replacer	✅
Field Splitter	✅
Field Type Converter	✅
Hive Metadata	✅
Schema Generator	✅
Stream Selector	✅
Field Pivoter	✅
Data Parser	✅

Executors: Stage name	Supported
Shell	⚠ MANTA doesn't support script analysis.

JSP 2.0 Expression Language (EL)

- Evaluation of the EL functions with runtime values (`str:trim` function included)
  - Runtime parameters (defined in the pipeline parameters)
  - Runtime properties (defined in `sdc.properties` or in a separate runtime properties file)
  - Runtime resources (defined file(s) in the `$SDC_RESOURCES` directory, by default `/opt/streamsets-datacollector/resources`; each file must contain one piece of information to be used when the resource is called)
- Evaluation of the EL functions, where the field path can be stored, to determine the field (column) dependencies in the Expression Evaluator stage

Unsupported Features

Here are some examples of StreamSets features that MANTA doesn't support yet.

- Evaluation of the extended regular expressions (Field Path Expression Syntax) that are used in field definitions
- Following features from the StreamSets Control Hub: topologies, fragments
- Evaluation of all JSP 2.0 Expression Language (EL) functions excluding the abovementioned supported features

Extraction of Jobs

As of MANTA Flow 1.34, job extraction from StreamSets Control Hub is supported. Pipeline extraction and job extraction can be independently turned on and off using the properties `streamsets.extractor.use.pipeline.extraction` and `streamsets.extractor.use.job.extraction`. If `streamsets.extractor.service` is set to "Data Collector", `streamsets.extractor.use.job.extraction` **must not** be set to true.

Each job is built on a pipeline, and job dataflow is typically similar to underlying pipeline dataflow. However, there can be differences caused by job configuration. It can affect the values of various node attributes, make the job connect to a different database, and change the names of tables and columns. In such cases, job dataflow better reflects reality. If the job is extracted, the extraction of the corresponding pipeline gives no additional information and slows down the analysis. It is therefore recommended, if possible, to use job extraction and disable pipeline extraction.

Kafka

⚠ Please note that this scanner is considered experimental, and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee the success of the connection or integration since we have no control, liability, or responsibility for third-party products or services, including for their performance.

MANTA can either connect to Confluent Platform Schema Registry and extract the schemas contained in Kafka topics in an automated way (Schema Registry Integration requirements), or it allows the user to describe the Kafka environment on their own (Manually Provided Schemas Integration Requirements) to visualize it and benefit from integrations with other scanners. The Kafka visualization includes objects such as a cluster, topics, schemas, and columns.

Supported Features

This is a list of the Kafka Schema Registry features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

- HTTP basic authentication for Schema Registry
- JSON schemas
- Avro schemas
- Integrations with DataStage and StreamSets scanners
- The option to manually define Kafka objects by providing a simple JSON file
- Extraction of schema from JSON payloads (as of R36)
- Informative attributes providing additional information about Kafka cluster (broker addresses, schema names, subjects in Schema Registry) (as of R36)

Unsupported Features

Here are some examples of Kafka Schema Registry features that MANTA doesn't support yet.

- Confluent Cloud Schema Registry and other Kafka distributions apart from Confluent on-premises
- Different subject name strategies (RecordName, TopicRecordName)
- Protobuf schemas
- REST API calls to Schema Registry over HTTPS
- Exports to third-party tools

Schema Registry Integration Requirements

The Schema Registry extractor connects to Schema Registry over REST API, extracts all possible schemas, and processes them to obtain information about their columns.

To achieve this, it is necessary to meet specific requirements and provide connection parameters as follows.

- Confluent Platform Schema Registry accessible via network
- The scanner is developed against Confluent Platform Schema Registry 11.0.8. Newer or older versions of Confluent Platform Schema Registry might be processed without an issue, but MANTA can't guarantee it.
- Connection parameters for Schema Registry
  - Schema Registry address
  - Schema Registry port
  - (Basic HTTP authentication activated) user
  - (Basic HTTP authentication activated) password
- TopicName naming strategy for subjects in Schema Registry

Manually Provided Schema Integration Requirements

Manually providing schemas is a way of visualizing the Kafka environment manually or semi-automatically by specifying which topics appear in which Kafka cluster and providing the schemas present in each topic (either by listing them or by schema file). The text below describes what format the metadata should be in, where it should be placed, and common mistakes associated with manual input.

Manually provided schemas must be placed in the `mantaflow/cli/input/kafka/<dictionary.id>` folder. The folder contains a file named `kafkaSchemaDefinitions.json`, which defines the Kafka objects in the Kafka cluster related to the `dictionary.id`.

`kafkaSchemaDefinitions.json` has to follow the prescribed structure.

```
{
   "cluster":"sample_cluster",
   "topics":[
      {
         "topicName":"topic1",
         "schemas":[
            {
               "schemaFile":"schema1.jschema",
               "fields":[
```

```
                  "column1",
                  "column2",
                  "column3"
               ]
            },
            {
               "id":1,
               "subject":"sample_subject",
               "version":1,
               "schemaFile":"schema2.avro"
            }
         ]
      },
      {
         "topicName":"topic2",
         "schemas":[
            {
               "fields":[
                  "column1",
                  "column2",
                  "column3"
               ]
            }
         ]
      }
   ]
}
```

White space characters are ignored.

Field Overview

Field	Description
cluster	Kafka cluster name such as `dictionary.id`
topics	JSON array of topics in the Kafka cluster
topicName	Kafka topic name
schemas	JSON array of schemas in the topic
schemaFile	Name of the schema file from which the fields (columns) will be obtained. The schema file must be in the same folder as the `kafkaSchemaDefinitions.json` file. Providing the `schemaFile` is optional, but note that at least one `schemaFile` or field should be provided for each schema. The schema files have to be encoded in UTF8. Scanner currently supports two types of schema files. › Avro schema—the .avro or .asvc file extension › JSON schema—the .jschema file extension › JSON file—the .json file extension
fields	JSON array of strings used to manually define fields. Providing the fields is optional, but note that at least one `schemaFile` or field should be provided for each schema.
id	Schema Registry schema ID. Providing the ID is optional.
subject	Schema Registry schema subject. Providing the subject is optional unless the version field is filled in.

version	Schema Registry schema version. Providing the versuib is optional unless the subject field is filled in.

Common Mistakes Associated with Manual Input Extraction

- Omitting some of the arbitrary fields (see above)
- Invalid parentheses
  - Note that schemas and topics are JSON arrays, and they need to start and end with square brackets
- The manual input definition file is placed in the wrong subfolder, not `mantaflow/cli/input/kafka/<dictionary.id>`
- Misnamed manual input definition file; it has to be `kafkaSchemaDefinitions.json`
- Schema files not put in the same folder
- Invalid file permissions for a manual definition file or referenced schema file
- Wrong file extension

Azure Data Factory

⚠ Please note that this scanner is considered experimental, and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee the success of the connection or integration since we have no control, liability, or responsibility for third-party products or services, including their performance.

- Azure Data Factory: Version 2 (V2)
- Azure Data Factory exports in JSON format contained inside a structured folder. The structure of the folder should match the folder structure one gets from downloading files from Git.
  - The structure:
    - Root folder
      - Dataflow (contains data flow resources)
      - Dataset (contains dataset resources)
      - Factory (contains a factory resource)
      - LinkedService (contains linked service resources)
      - Pipeline (contains pipeline resources) *(Unsupported)*
- Automatic extraction is not supported. Files have to be exported manually from ADF. For more information, see https://mantatools.atlassian.net/wiki/spaces/MTKB/pages/748421302/MANTA+Flow+Usage+Preparing+Scanner+Inputs#Azure-Data-Factory.

Supported Features

This is a list of the ADF features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

Legend

- ✓ Fully supported
- ✗ Not supported (Can cause missing lineage)
- ? Not supported (Can't cause missing lineage)

Supported Resources

Resource type	Supported?
Factory	✓
Pipeline	✗
Data flow	✓
Flowlet	✗
Dataset	✓
Linked service	✓

Data Flow Transformations

Transformation name	Supported?
Aggregate	✗
Alter row	?
Assert	?
Conditional split	✓
Derived column	✓ *(Column patterns are not supported)*
External call	✗
Exists	✗
Filter	✓
Flatten	✗
Join	✓
Lookup	✓
New branch	✓
Parse	✗
Pivot	✗
Rank	✗
Select	✓ *(Rule-based mappings are not supported)*
Sink	✓
Sort	?
Source	✓
Stringify	✗
Surrogate key	✓
Union	✓
Unpivot	✗
Window	✗

Examples of Unsupported Features

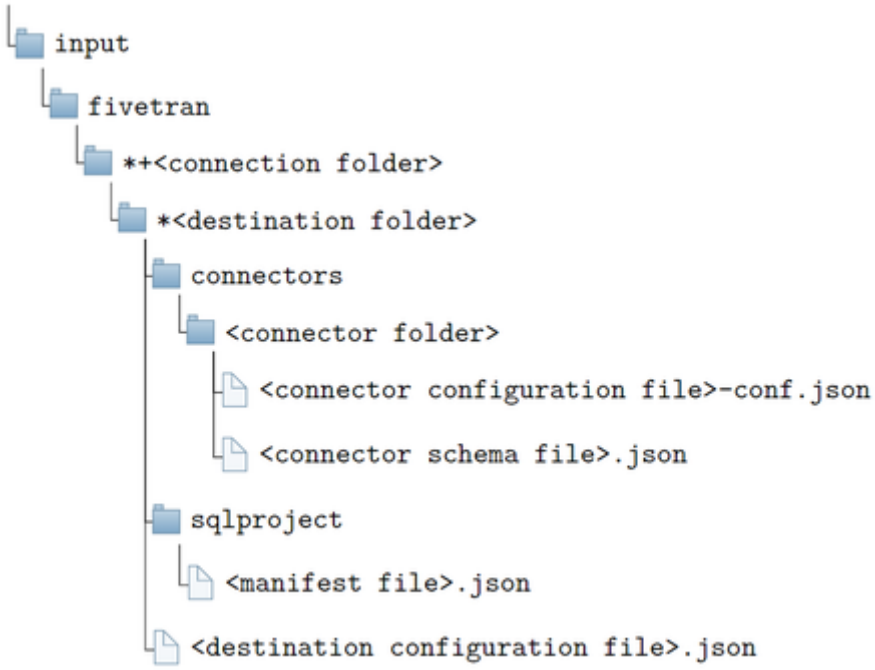
- The folder structure of ADF project visible in ADF UI is not represented in the ADF GIT repository or in MANTA. This leads to all files of the same resource type being located in the same folder (dataflow, linkedService, etc.) in MANTA Repository.
- Expressions are not evaluated. This leads to the following.
  - SQL queries constructed via expressions cannot be analyzed.
  - References to drifted columns with the expression `byName(<column name string>)` cannot be analyzed.
  - References to columns with the expressions `$$, $#, $0` cannot be analyzed.
- If the original schema is not saved in the dataset, drifted columns might not be shown in the lineage. For more information on schema drift, see https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-schema-drift.
- Rule-based mappings of select transformations and column patterns of derived column transformations are not supported.

Fivetran

⚠ Please note that this scanner is considered experimental, and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee the success of the connection or integration since we have no control, liability, or responsibility for third-party products or services, including their performance.

- The scanner has been developed to accommodate the September 2022 version of Fivetran. It aims to support. Newer versions of Fivetran might be processed without an issue, but MANTA can't guarantee it.
- The Fivetran scanner processes files in JSON format contained inside a structured folder that needs to be manually provided.
  - The structure:



    - Legend:
      - Parts of folder and file names set off by **<>** can be changed.
      - Folder and file names marked with a ***** are used in the visualization.
      - Folder and file names marked with a **+** are determined by configurations.
    - Description:
      - Scanner folder (*fivetran*)
        - The name of this folder can't be changed.
        - The folder can contain multiple Fivetran connections.
      - Connection folder (*connection folder*)
        - The name of this folder must be the same as the *Fivetran Server Name* in Admin UI.
      - Destination folder (*destination folder*)
        - The folder must contain a configuration file (*destination configuration file*).
      - Folder with connector folders (*connectors*)
        - The folder can contain any number of connectors.
      - Folder for one connector (connector)
        - The folder must contain two files.
          - A file containing a connector schema (*connector schema file*)
          - A file containing a connector configuration (*connector configuration file*)
      - Folder with the SQL project (*sqlproject*)
        - The folder should contain:
          - Exactly one manifest file from the dbt transformation project
          - Multiple SQL files from the basic SQL project with SQL extensions
          - Nothing if no transformation project is used
- Automatic extraction is not supported yet. Files have to be exported manually from Fivetran or dbt. For more information, see MANTA Flow Usage: Preparing Scanner Inputs.

Fivetran Scanner Overview

The Fivetran scanner analyzes destinations, database connectors, and the objects they contain. If the Fivetran scanner and relevant database scanners are configured properly, MANTA Viewer can display lineage starting with source databases, flowing through connectors showing their naming conversions, and then right into a destination database. The lineage of Basic or dbt transformations and their connections to the destination database can be displayed, too.

Supported Features

This is a list of the Fivetran and dbt features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

In order for all supported features to work properly, the connectors must be synchronized and a transformation project executed at least once.

For all connector and destination databases to be processed, there should be a connection created in a relevant scanner.

Legend

- ✓ Supported
- ✗ Not supported (and can cause missing lineage)

Connector types	Supported
Database Connectors	✓ (Only for databases supported by MANTA)
Application Connectors	✗
File Connectors	✗
Event Connectors	✗
Function Connectors	✗

Connector features	Supported
Column Blocking	✓
Column Hashing	✓
Schema Name Conversions	✓
Table Name Conversions	✓
Column Name Conversion	✓ (Transliteration isn’t supported)

Transformation projects	Supported
Basic SQL Transformations	✓
Transformations for dbt Core	✓

dbt features	Supported
Pre and post-hooks	✗
on-run-start and on-run-end	✗
Macros	✓
Materializations	✓
Aliases	✓
Custom schemas	✓

BI and Reporting Tools

IBM Cognos

Note
Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- The scanner is developed against Cognos Analytics 11.0.8. It aims to support Cognos Business Intelligence version 10 and newer and Cognos Analytics until 11.0.8. Newer versions of Cognos Analytics might be processed without an issue, but MANTA can't guarantee it.
- Cognos Analytics: version 11 and newer
- Java 11 is not required but is strongly recommended, especially if you run into issues when the dataflow analysis is stuck
- Certain Cognos SDK libraries that match the version of the Cognos server; these libraries are described further in the section below
- User having the following roles:
  - *Directory Administrator* with read permission (ensures that all reports will be available for extraction)
- Connection parameters:
  - URL of the dispatcher service
  - Anonymous access indication or all of the following
    - User name
    - User password
    - ID of the namespace that the user belongs to
- Server must be accessible via network and running
- To handle many large Framework Manager models (FMD project files of about 100MB or more), it might be necessary to increase the maximum heap space for a JVM, both on the Cognos side and the MANTA side.
  - For the MANTA side, this can be done in Process Manager as described in MANTA Process Manager#Scenario-Execution-Runtime-Limits. Depending on the size of the Framework Manager model, 10 or even 15GB may be needed.
  - This Cognos page describes how to increase allocated memory in Cognos.
- To process Dynamic Cubes and the related data flows, Cube Designer project files in FMD format need to be provided manually and the manual mapping file `searchPathMappingManual.csv` describing the relations between package search paths and file locations needs to be edited accordingly. The Dynamic Cube configuration for the Cognos scanner is described in detail on the page Cognos Resource Configuration.
  - The manual mapping also requires certain knowledge of Dynamic Cube usage in the company and the ability to determine which Dynamic Cubes are used in some reports in Cognos Analytics.

Cognos SDK Libraries

MANTA cannot distribute Cognos SDK libraries because they are owned by IBM and licensed accordingly. However, Cognos SDK and its libraries are usually included in the contract when a Cognos Analytics software license is purchased / subscribed to by a customer. Copy the following libraries

- `axis.jar`
- `axisCognosClient.jar`
- `jaxrpc.jar`

from <COGNOS_SDK_INSTALL_DIRECTORY>/sdk/java to `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/scenarios/manta-dataflow-cli/lib-ext` (`<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/scenarios/manta-dataflow-cli/lib` in versions prior to 1.32), restart the MANTA Service Utility, and be sure to reboot MANTA Apache Services when finished.

NOTE: These files are associated with the dispatcher component of Cognos (as opposed to the Content Manager component).

Cognos Scanner Overview

The Cognos scanner extracts XML specifications of reports, interactive reports, data modules, Framework Manager models, and data sets from the content store as well as database connection information. It analyzes reports and all the objects that the reports depend on and produces data lineage. If the Cognos scanner and relevant database scanners are configured properly, MANTA Viewer can display lineage starting with databases or uploaded files, flowing through Framework Manager models or data modules right into the report's query objects, and finally to the visualization elements called report items.

Supported Features

Note

This is a list of Cognos features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Other than that, features that are not listed are primarily considered not supported.

- Framework Manager models (including multilingual definitions)
- Relational reports dependent on a Framework Manager model
- Report queries (including SQL-based queries and queries created using operations such as joins and unions)
- Data modules consisting of data from:
  - Databases
  - Other data modules
  - Data sets
  - Uploaded files such as XSLX, CSV, and others
- Reports dependent on a data module
- Database connection support, even reports with SQL placed directly in query objects
- Almost all report visualizations and inner objects such as queries
- Filters in report queries
- Publishing expressions (calculations) that are used to calculate various data items
- Multilingual Framework Manager models: Please note that objects in Cognos reports are NOT multilingual; their names are unique and can be set in edit mode. Report items can have localized labels, but these labels can't be easily matched to the relevant report item, which is why non-localized technical names are used as object identifiers in MANTA viewer.
- Dynamic Cubes provided as Cube Designer projects (including multilingual definitions and virtual cubes)
- Dynamic Cubes defined in a stand-alone Cube Designer project and referenced via Framework Manager models (as of MANTA 1.34)

Examples of Unsupported Features

Note
These are some examples of Cognos features that MANTA doesn't support yet. Please note that this list is not complete. We plan to support most of the listed features in future releases.

- Multilingual reports
- Dynamic Cubes designed in Framework Manager as part of a DQM package
- Data sources based on Power Cubes, TM1 Cubes. Note that Cognos Finance, Cognos Now, and Cognos planning data sources are not even supported by Cognos Analytics itself as of 11.0.4.
- Dashboards and stories
- Map visualizations
- Reports using a lot of advanced query macros and prompts might not be accurate, as they use run-time information that MANTA can't analyze

Microsoft Excel

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Excel workbooks in XLSX or XLSM format
- ODC files (required if a PowerPivot data model is used)
  - Located in the `C:\Users\<username>\Documents\My Data Sources` directory by default in Windows
- The scanner is designed to analyze and import data lineage information to MANTA based on formulas, pivot charts and pivot tables, and connections to databases and other sources. It is not intended for importing actual data/rows to MANTA nor does it do so.

Microsoft Power BI

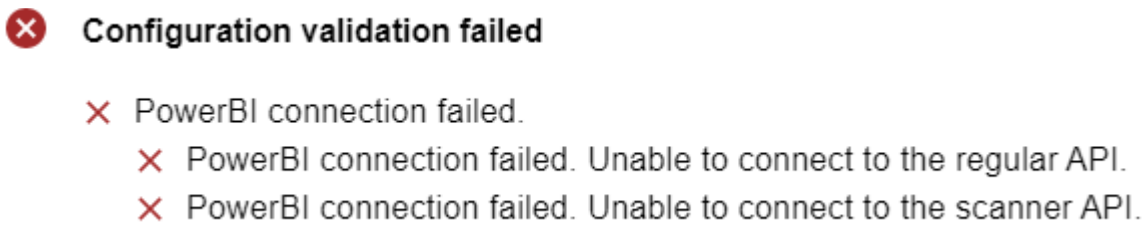
The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Power BI Pro
- Power BI Premium
- Power BI Report Server

MANTA supports two extraction modes with different approaches to extracting report definitions: Azure and local. If it’s possible, please use Azure mode, as it provides much better data lineage. Each mode has different requirements and limitations as described below.

Azure Extraction Mode

Azure extraction mode allows automatic extraction using a REST API to connect to Power BI Pro/Premium through Azure Active Directory. After creating a new empty Power BI connection in MANTA and clicking on the *Validate* button, you should see the following.



It's necessary to correctly set up access to these two API's.

- First, set up access to the regular API following this guide - How to Set Up Power BI Regular API.
- Then, set up access to the scanner API following this guide - How to Set Up Power BI Scanner API.

Scanner API

Microsoft has released an enhancement of their Power BI API which newly provides metadata that were previously unavailable. We have reworked and updated our MANTA Power BI scanner to use this new Admin API. The benefits of this update are huge. First, our scanner can now generate correct data lineage for measures and calculated columns. Second, it's no longer necessary to manually input the Power Query scripts, as they're now automatically imported using the new API. Some other smaller improvements have also been made. Find more information about Scanner API here:

- https://powerbi.microsoft.com/en-us/blog/announcing-new-admin-apis-and-service-principal-authentication-to-make-for-better-tenant-metadata-scanning/
- https://powerbi.microsoft.com/en-us/blog/announcing-scanner-api-admin-rest-apis-enhancements-to-include-dataset-tables-columns-measures-dax-expressions-and-mashup-queries/

You can use Scanner API to generate report lineage under one condition—the report's data set must be in a Power BI data set cache. Whenever a data set is refreshed or any report containing the data set is published, then that data set is added to the data set cache for 30 days. However, if the data set is not in the cache, Scanner API doesn't provide the low-level metadata which is necessary to generate the correct data lineage. In such cases, the `cannot_use_scanner_api_for_report` error is shown in the log and the old regular API is used to generate data lineage. This can be fixed by manually refreshing that data set in the Power BI app or by republishing any report with this data set. We plan to automatically refresh that data set in such a situation in the future.

To summarize, Scanner API provides huge benefits under one condition—the scanned reports must be refreshed or republished in the last 30 days. Automatic refresh can be configured using this official guide https://docs.microsoft.com/en-us/power-bi/connect-data/refresh-scheduled-refresh.

Local Extraction Mode

Local extraction mode allows automatic extraction using a REST API to connect to Power BI Report Server.

- User permission requirements
  - Read content
  - Read report definitions
  - Read properties
- Connection requirements
  - Protocol used to connect to the server (for example, HTTP)
  - IP address of the server
  - Port on which the server listens for connections
  - Virtual directory on the server (for example, http://192.168.0.16:80/**BIReports**)
  - User name
  - Password

Supported Features

This is a list of Power BI features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Other than that, features that are not listed are primarily considered not supported.

- Reports consisting of data from:
  - Databases
  - Uploaded files such as XSLX, CSV, and others
- Reports in .PBIX format
  - Report visualizations including these features
    - Slicers
    - Measures (as of R35) *
    - Calculated columns (as of R35) *

    - Aggregations and data hierarchies
  - Metadata visualizations
    - Power Query scripts
    - Endorsement level, DAX formulas, author, last modification, etc. (as of R37) *
  - Automatic extraction of Power Query scripts (as of R35) *
    - A manual workaround if you are not using Scanner API as described in How to Manually Insert Power Query Script
- Paginated reports (as of R37.1)
  - Support for RDL files: paginated reports
  - Support for RDS files: shared data sources
  - Support for RSD files: shared datasets

*These features are only supported in the Azure extraction mode when access to Scanner API is set up correctly.

Examples of Unsupported Features

This is a list of Power BI features that MANTA doesn't support yet. Please note that this list is probably not complete, as there might be Power BI features that we don't know of. Most of the features listed below will be supported in future releases.

- Reports consisting of data from SSAS
- Reports that cannot be downloaded from the Power BI Azure application (i.e., reports that have the option `Download the .pbix file` grayed out in the UI, even if you have the correct rights), which include:
  - Reports created in PowerBI Desktop that were created together with a dataset which is now in the large dataset storage format
  - You can find all such exceptions in the official Power BI documentation https://docs.microsoft.com/en-us/power-bi/create-reports/service-export-to-pbix#limitations
- Unsupported use cases of paginated reports
  - Paginated reports based on a Power BI Azure shared dataset
  - Mobile reports
  - KPIs
- Windows authentication in local extraction mode

Feedback

If you have any feedback regarding the implementation of the new Scanner API or any feature suggestions, we'd love to hear from you. Please contact our support team at http://portal.getmanta.com.

Microsoft SSAS

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- SQL Server Analysis Services (SSAS): version 2012 or newer, tabular models up to and including version 1400—newer tabular models might not work (Azure Analysis Services, most SQL Server 2017 instances; see the SQL Server documentation)
- Server must be in **SSAS Tabular Model** or **SSAS Multidimensional Model** mode
- Able to obtain the database source as an XML file
  - By exporting the database using Microsoft SQL Server Management Studio
    - Log in as a user who is a member of a group with ***Full Control (Administrator)*** permissions for the database
    - Right click on *SSAS Database* in the object explorer
    - Select *Script Script Database As CREATE To File*
    - Save the export as an ***.xmla** file
  - By copying the source file of the model from Microsoft Visual Studio
    - Find the *Analysis Services Tabular Model* file in the *Solution Explorer* window and click on it
    - Check the properties of the file and obtain the full path
    - Copy the ***.bim** file specified by the path

SSAS Scanner Overview

Because the SSAS scanner requires the user to manually obtain database sources, it is necessary to move the input files to a folder whose structure matches that of the actual SSAS server and database environment.

The SSAS server is one level above a database. It is possible to have multiple SSAS databases inside a single SSAS server. This structure can be seen in SQL Server Management Studio. It also corresponds to the way exporting data to an `.xmla` file is described above. You select one database (out of many) inside the server for export, and it produces a single `.xmla` file that is only for the selected DB. It is necessary to move the file to a folder with a structure that matches the one inside the SSAS server. So, for example, avoid putting databases with the same name in the same input directory—it would be the same as having two databases with the same name inside a single server. To summarize, SSAS server = input folder, SSAS database = `.xmla` file.

It is not necessary to have one connection per SSAS input file. If it matches the real environment, then it should be safe and more intuitive to share the same connection for all databases on a single server. But especially during initial testing, it is common to see some testing databases using the same dummy name inside a single input folder (e.g., different versions of the same DB testing different features) that produces incorrect output. In the latter case, it is safer to use one connection per input file.

Input Example

For this example, consider an input folder representing the SSAS server that contains two database files: one as `Model.bim` and the other as an `Example.xmla` file.

If the database name in `Model.bim` is different than the one in `Example.xmla`, the analysis will process them correctly.

If both files represent the same database (databases with the same name), this will cause the previously described error—having two databases with the same name inside a single server. The database name is not given by the file name or suffix; it is specified inside the source file. The database file's name does not actually have any meaning and won't influence the lineage, but it is recommended to use the name of the database inside to clearly distinguish what database it represents.

If both files actually represent the exact same database, only one needs to be used as analysis input (it doesn't matter which one); for example, place only `Example.xmla` in the input folder.

If you need to analyze the databases from both the `Model.bim` and `Example.xmla` files (e.g., if they are different versions of the same database), it is necessary to place each file into its own separate folder.

Supported Features

ℹ This is a list of SSAS features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Other than that, features that are not listed are primarily considered not supported.

- Most features showcased in https://learn.microsoft.com/en-us/analysis-services/analysis-services-tutorials-ssas
- Data source definitions using:
  - SQL
  - M language (PowerQuery)

Examples of Unsupported Features

ℹ This is a list of SSAS features that MANTA doesn't support yet. Please note that this list is probably not complete, as there might be features that we don't know of. Most of the features listed below will be supported in future releases.

- MDX:
  - Member identifiers containing four parts: [Cube].[Dimension].[Level / Hierarchy].[Member] (expected by R39)
  - Variables
  - Functions listed in https://learn.microsoft.com/en-us/dax/table-manipulation-functions-dax
  - Parameterized MDX queries (example in https://learn.microsoft.com/en-us/analysis-services/multidimensional-models/mdx/using-variables-and-parameters-mdx)
- DMX (https://learn.microsoft.com/en-us/sql/dmx/data-mining-extensions-dmx-reference)
- Deduction of objects caused by incomplete input
- Using KPI as a data source for another technology (like SSRS)

Microsoft SSRS

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- SQL Server Reporting Services: version 2010 or newer
  - Only paginated reports are supported (mobile reports are ignored)

MANTA supports two extraction modes with different approaches to extracting report definitions: REST and DATABASE. Each mode has different requirements and limitations as described below.

REST Extraction Mode

- Preferred extraction mode that allows automatic extraction of all required information, using REST API to connect to the SSRS server
- Can be used with SSRS version 2017 or newer
- Report Server must be configured to use NTLM authentication
  - Authentication configuration details are described here

  - Disabling other authentication types may be required; for example, if negotiate authentication is also enabled, the extraction may not be able to connect to Report Server
- Requires a user with a login for SSRS Web Services who is able to connect using NTLM authentication and who must be assigned the following roles.
  - *Report Builder*
    - Allows the user to view report definitions
    - Required for report extraction
  - *Publisher*
    - Allows the user to publish reports
    - Required for data sources
      - They are used to extract report connection information (which databases are used as data sources for the given report)
    - A user without this role cannot see any data source definitions
    - May be assigned only for concrete data source objects, not for a whole server to minimize user privileges
- Alternatively, the user can have just one role, the more privileged role of *Content Manager*, to be able to download all the information
- Connection requirements
  - SSRS Web Service must be accessible via TCP
  - Name or IP address of the SSRS Web Service
  - Port on which the SSRS Web Service listens for connections
  - Protocol used to connect to SSRS Web Service (for example: HTTP)
  - User name
  - Password

DATABASE Extraction Mode

- Use a JDBC connection to the MS SQL database where the published reports are stored
- Can be used with SSRS version 2010 or newer
- Extraction may download out-of-date information for data source definitions if after publication they were modified using SSRS Web GUI
- Requires a user with SQL Server authentication and privileges for SELECT on the table `dbo.catalog`
- Connection requirements
  - SQL Server must be accessible via TCP
  - Name or IP address of the SQL Server database where SSRS is deployed
  - Port on which the SQL Server database listens for JDBC connections
  - User name
  - Password
  - Name of the catalog where SSRS reports are stored
  - MANTA supports the following authentication types for MS SQL server:
    SQL Server
    The user credentials are configured in the MS SQL server. SQL Server stores both the username and a hash of the password in the master database using internal authentication methods to verify login attempts. To use this type, fill in the username and password.
    Native (Windows Only)
    The identity of the user running MANTA is used to authorize access to the MS SQL server. MS SQL delegates the authentication to the Windows client where the user logged in.
    To use native authentication, the property `integratedSecurity=true` has to be set in the JDBC connection string. The Configurator adds this property automatically when the connection is saved.
    NTLM (as of MANTA Flow R34)
    NTLM authentication allows you to authenticate a user against an active directory. To use NTLM authentication, you can, in addition to the username and password, fill in the domain property that identifies the domain controller.
    To use NTLM authentication, the properties `authenticationScheme=NTLM;integratedSecurity=true` have to be set in the JDBC connection string. Configurator adds those properties automatically when the connection is saved.

OBIEE

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- The scanner has been developed for Oracle Business Intelligence 12.2.1.4.0. It aims to support Oracle Business Intelligence version 12. Other versions might be processed without an issue, but MANTA can't guarantee it.
- The memory allocation for MANTA server must be 10GB or more.
- To be able to analyze data models, reports, and dashboards, the following is required:
  - The user* must be granted *Open* permission for every file they want to analyze.
  - The user must be an *Administrator* to automatically link data flows to data sources.
  - Connection parameters:
    - URL of OBIEE Server
    - User name
      - The user can be created in the WebLogic Server Administration Console as follows: *Security Realms* section choose a realm *Users and Groups* tab create a new user

    - User password
  - The server must be running and accessible via network.

Manual Extraction

If you want to analyze only data models, reports, and dashboards, you can skip the manual extraction step. However, if you want to analyze analyses and repository files, you need to manually extract the following two files.

- Repository dump—How to Manually Extract the Repository Dump in OBIEE
- Catalog Manager export—How to Manually Extract the Catalog Manager Export in OBIEE

OBIEE Scanner Overview

The OBIEE scanner extracts XML specifications for reports, data models, analyses, and dashboards from the presentation catalog as well as database connection information. It analyzes reports, analyses, repositories (RPD files), and all the objects they depend on and produces data lineage. If the OBIEE scanner and relevant database scanners are configured properly, MANTA Viewer can display lineage starting at the databases and flowing through the data models' data sets right to the report's components. For analyses, the OBIEE scanner can also display lineage starting at the databases, flowing through the physical, logical, and presentation layers of the repository, and ending with analyses.

Supported Features

This is a list of OBIEE features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Other than that, features that are not listed are primarily considered not supported.

- Data models
  - Data models with the following data set types
    - SQL query
    - Oracle BI analysis
    - Web service
    - CSV file
    - Microsoft Excel file
    - XML file
  - All types of value sets from the list of values
    - SQL query
    - Fixed data
  - All types of data model parameters
    - Menu
    - Date
    - Text
  - Expressions in data sets—partially
- Reports
  - The following types of data sources for reports
    - Data model
    - Spreadsheet
    - Subject area—partially
  - BI Publisher layouts
  - All report components of the BI Publisher layout
    - Layout grid
    - Data table
    - Chart
    - Pivot table
    - List
    - Repeating section
    - Text item
    - Gauge
    - Image
  - Formulas and filters in report components—partially
- Repository (RPD) files
  - Physical, logical, and presentation layers
  - Connections between layers
  - Connections between the physical layer and underlying databases
- Analyses
  - The following types of data sources for analyses
    - Subject area
    - Simple logical SQL data sources
  - Analysis criteria

    - Regular columns
    - Formulas
    - Filters—partially
  - The most important analysis views
    - Title
    - Table
    - Pivot table
    - Performance tile
    - Treemap
    - Heat matrix
    - Trellis
    - Graph
    - Gauge
    - Funnel
- Dashboards with the following views
  - Analysis
  - Report

Examples of Unsupported Features

This is a list of OBIEE features that MANTA doesn't support yet. Please note that this list is probably not complete, as there might be OBIEE features that we don't know of. Most of the features listed below will be supported in future releases.

- Data models
  - Data models with LDAP query, MDX query, HTTP, view object, and content server data sources
  - Event triggers, flexfields, and bursting in data models
- Repository files
  - Variables, dimensions, and hierarchies
- Analyses
  - Analyses with database query—created by “Create direct database query”
  - Maps, filters, selection steps, and other views in analyses
  - Analysis prompts
- Dashboards
  - Prompt, filter, and condition objects

Tableau

Please note that this scanner is considered experimental and MANTA only recommends using it for testing purposes.

The following are the prerequisites necessary for MANTA to connect to the Tableau third-party system, which you may choose to do at your sole discretion.

Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

The scanner can automatically extract workbooks and data sources from Tableau Online and Tableau Server instances or they can be extracted manually and put into the directory where MANTA expects to find the files for lineage analysis.

The following requirements must be met for the scanner to automatically to extract the data.

- Tableau Online or Tableau Server accessible from the internet
  - Versions as of 10.3 (May 2017) are supported
- Tableau REST API is enabled
- Connection parameters
  - URL of the Tableau Server
  - User name
  - User password
- A user with the appropriate permissions (see below)

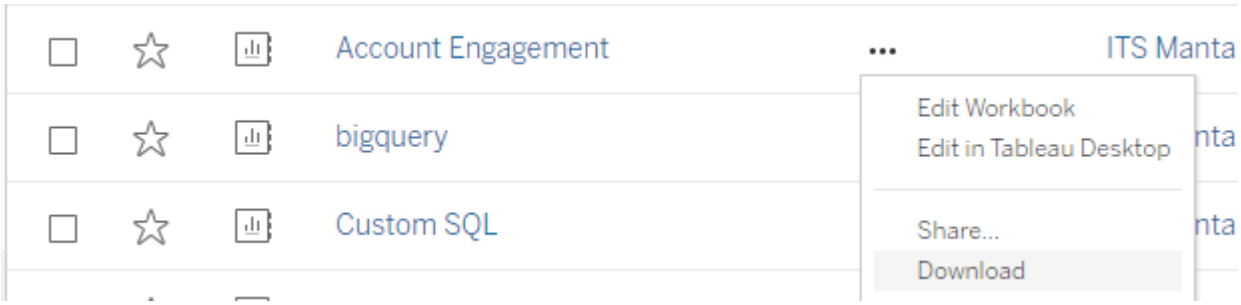
Providing Input Data Manually

The following requirements must be met to process files provided manually.

- The files are placed in `$MANTA_DIR/cli/temp/tableau/{sitename}/{projectname}` where the `sitename` is the name of the site (set in the MANTA connection) and the `projectname` is the name of the project the workbook or data source is in.

- The site name should match the name of the original site from which the items were downloaded. Otherwise, shared data sources may not work.
- Workbooks are downloaded as TWB or TWBX.
- Shared ("published") data sources are downloaded as TDS or TDSX.
- A project can be nested inside another project (projectname1/projectname2), but there has to be at least one project.

To download a workbook or data source from Tableau Server, locate it inside its parent project, click on the three dots next to its name, and select *Download*. (Alternatively, open—view, not edit—the workbook and click on the three dots next to its name.)

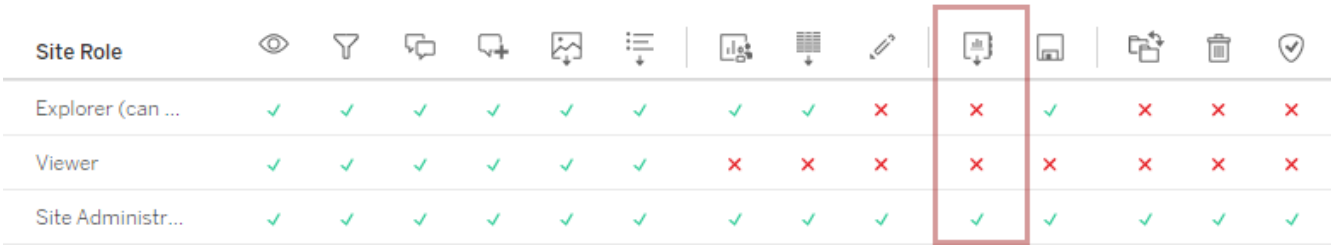


It's also possible to serve workbooks created on Tableau Desktop that are not part of any site on any server. Create the same file structure as described above (.../`tableau/{sitename}/{projectname}`) with any site name and any project name.

Permissions

To download workbooks and data sources, MANTA needs the credentials of a user with access to them. The user must have one of the following roles.

- *Explorer* (can publish) or higher—users with roles lower than *Site Administrator* must have permission to download each workbook and data source to be analyzed (this setting has to be applied to each workbook separately)



- *Site Administrator*, which always has permission to download everything

Supported Features

This is a list of Tableau features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

- Workbooks
  - Worksheets, dashboards, stories
  - Row and column shelves
  - Properties of the chart set on the marks card (color, angle, size, etc.)
  - Filters and pages
  - “Measure Values” and “Measure Names”
- Data sources
  - Published data sources as well as data sources that are part of a workbook
  - Connection to a database (PostgreSQL, Oracle, MS SQL, DB2, etc.)
    - Either using custom SQL or loading the entire data set
  - Connection to a file (CSV, XLSX, etc.)
  - User-defined calculations
  - Sets
  - Groups
  - Parameters (not SQL, see below)
  - Geographical data
  - Extracts
  - Joins
  - Union
  - Pivot

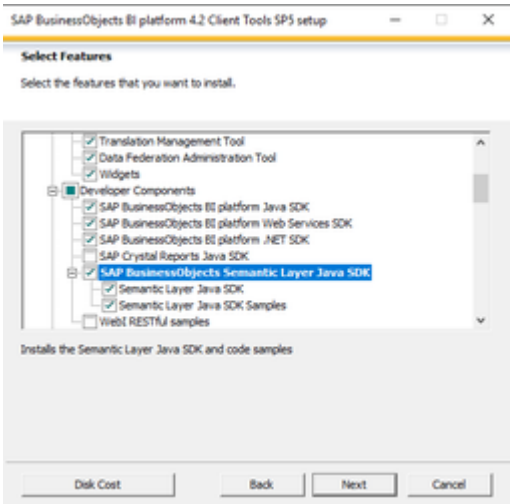
Known Limitations

- The scanner is able to process workbooks created in Tableau version 10.3 (May 2017) and newer. Older workbooks may work as well, but an error might appear in the analysis.
- Tableau supports “parameters”, which can be put into an SQL query. MANTA is able to visualize parameters, but not as part of SQL.
- Initial SQL is not supported.
- Connections to Hive are not supported yet.
- There is a known issue with connecting to two or more sheets from one Excel file.
- Tableau Prep is not supported.
- Pages on dashboards are not supported.
- Tableau 2019.1 and older are supported but the REST API version property has to be configured.
- Numeric expressions in scientific notation (e.g., 2E-3) are not supported in calculated columns.
- Reference and intersection filters are not supported.
- *Measure Names* and *Measure Values* columns are supported. It is also possible to select only some columns to be part of the result using a filter. However, only the *Include Values* filter (default type) is supported, not *Exclude Values*.
- Relationships with more than one column in the expression are not supported yet. Relationships that use calculated columns in an expression are not supported either.
- Workbooks and data sources in the user's personal space won't be extracted.
- Connections using stored procedures are not supported.
- IN logical function is not supported.

SAP BusinessObjects

Note
Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- The scanner is developed for SAP BusinessObjects Enterprise XI 4.0. It aims to support SAP BusinessObjects XI 4.0. Newer BO server versions, but not newer than 4.3, might be processed without an issue, but MANTA can't guarantee it. Version 4.3 is not supported.
- Client Tools 4.0 with *Semantic Layer Java SDK* installed (newer versions of SAP BO Client are not supported)
  - Client Tools can only be installed in Windows; Linux is not supported
  - Make sure this option is checked in the Client Tools installation, as shown in the picture below
    - 

  - Required JAR files:
    - In the folder `${SAP_BO_INSTALLATION_FOLDER}/SL SDK/eclipse/plugins`
      - `com.sap.sl.sdk.authoring.jar`
      - `com.sap.sl.sdk.framework.jar`
    - In the folder `${SAP_BO_INSTALLATION_FOLDER}/java/lib`
      - `aspectjrt.jar`
      - `bcm.jar`
      - `ccis.jar`
      - `ceaspect.jar`
      - `cecore.jar`

    - `celib.jar`
    - `cesession.jar`
    - `ConnectionServer.jar`
    - `corbaidl.jar`
    - `cryptojFIPS.jar`
    - `dsl_engine.jar`
    - `ebus405.jar`
    - `i18n4j.jar`
    - `jcmFIPS.jar`
    - `logging.jar`
    - `procWebiPublishing.jar`
    - `sdk.core.jar`
    - `TraceLog.jar`
    - `XcelsiusDSL.jar`
    - `XcelsiusServer.jar`
    - `xpp3.jar`
  - In the folder `${SAP_BO_INSTALLATION_FOLDER}/java/lib`. The following libraries were used during development. Slightly older or newer versions should work, but MANTA cannot guarantee it.
    - `org.eclipse.emf.common_2.4.0.v200902171115.jar`
    - `org.eclipse.emf.ecore.xmi_2.4.1.v200902171115.jar`
    - `org.eclipse.emf.ecore_2.4.2.v200902171115.jar`
    - `org.eclipse.equinox.common_3.4.0.v20080421-2006.jar`
    - `org.eclipse.equinox.registry_3.4.0.v20080516-0950.jar`
    - `org.eclipse.osgi_3.4.3.R34x_v20081215-1030.jar`
- Other dependencies:
  - Accessible folder with 32-bit JRE installed with SAP BO at `${SAPBO_INSTALLATION_PATH}/win32_x86/jre8/bin/java`
  - Accessible BO connectivity directory at `${SAPBO_INSTALLATION_PATH}/dataAccess/connectionServer`
- User having the following roles:
  - Universe designer users with full control permission to the folders where documents and universes are stored
  - With full control permissions, it is possible to manually remove write access. Be aware that not setting full control and adding only read permissions is insufficient.
- Server must be accessible via network and running
  - Server must support RESTful API (HTTP port must exist and be open)

SAP BO Scanner Overview

The SAP BO scanner extracts the JSON/XML specifications of documents, universes, and data foundations as well as database connection information. It analyzes documents and all the objects that the documents depend on and produces data lineage. If the SAP BO scanner and relevant database scanners are configured properly, MANTA Viewer can display lineage starting with databases, flowing through data foundations and universes, right into the data providers or document variables, and finally to the visualization elements called report elements.

Extractor Usage

Extractor Not Fully Functional in R36
The extractor is fully functional in R35 and earlier versions of MANTA. In R36, the extractor only handles the automatic part of the extraction. For the manual part of the extraction, an earlier version of MANTA (R35 and earlier) must be used.

This limitation is caused by SAP BO—the manual part of the extraction (the part utilizing client tools) requires Java 8, and Java 8 is no longer supported in MANTA R36 and newer.

The current workaround requires two MANTA instances.

1. An R35 (or older) instance for extraction. (Required for the manual part, but can be used for the full extraction.)
2. An R36 (or newer) instance for dataflow analysis and, optionally, the automatic part of the extraction.

There are two extraction scenarios that can be followed.

1. If MANTA is installed on the same machine as SAP BO Client Tools, a full extraction can be executed on the machine—in this case, the parameter `sapbo.installation.path` must be filled.
2. If any of the previously mentioned requirements are not met, the extraction must be split into two parts.
   a. Running the automatic extraction on the main machine (the machine without Client Tools); in which case the parameter `sapbo.installation.path` should be left empty
   b. Running the manual extraction on the other machine (the machine with Client Tools); in which case it is necessary to fill in `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/scenarios/manta-dataflow-cli/bin/sapboExtractorSdkConfig.bat` and run `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/scenarios/manta-dataflow-cli/bin/sapboExtractorSdk.bat`. The output from this extraction (located in `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/temp/sapbo/${SAPBO_EXTRACTOR_ID}`) shall be merged with the output from the first extraction, located in the same folder but on the other machine.

Detailed Explanation of the Extraction Scenarios Mentioned Above

2.a: Windows scenario

1. Make sure both MANTA and SAP BO Client Tools are installed on the machine—this means it is a Windows machine.
2. Fill in all the parameters (including `sapbo.installation.path`).
3. Run the extraction.
   - The extracted folder, located in `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/temp/sapbo/${SAPBO_EXTRACTOR_ID}` after the first successful extraction, contains three folders required for further analysis.
     - Connections
     - Documents
     - Universes

2.b: Linux scenario or case where MANTA is **NOT** installed on the same machine as SAP BO Client Tools

1. Fill in all the parameters (excluding `sapbo.installation.path`) and run the automatic extraction.
   - If the parameter `sapbo.installation.path` is left empty, the *Connections* folder will be missing from the extraction result folder—this is a sign of an incomplete extraction, where only the automatic extraction ran.
   - The automatic extraction (without the parameter `sapbo.installation.path` filled in) only produces the folders *Documents* and *Universes* (there are still some more files missing from this folder).
2. Install MANTA on a machine that has SAP BO Client Tools installed.
   a. Fill in all the parameters in `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/scenarios/manta-dataflow-cli/bin/sapboExtractorSdkConfig.bat`
   b. Run the manual extraction— `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/scenarios/manta-dataflow-cli/bin/sapboExtractorSdk.bat`
      - The manual extraction produces the folders *Connections* and *Universes*.
      - It is necessary to merge these two folders produced by manual extraction into the folder produced by automatic extraction so that the folder structure fits the structure as listed above in the first point.
3. Copy the output from the manual extraction (the *Connections* and *Universes* folders from `<MANTAFLOW_INSTALLATION_DIRECTORY>/cli/temp/sapbo/${SAPBO_EXTRACTOR_ID}`) to the same folder (merge the folders) but on a Linux machine.

> **Note**
> It is also possible to run the whole extraction on one machine and copy its output on another machine, running the dataflow analysis there afterward. MANTA needs to be installed on both of these machines in this scenario.

Supported Features

> **Note**
> This is a list of SAP BO features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Other than that, features that are not listed are primarily considered not supported.

- Documents using UNX universes as their data provider
- Single-source data foundations
- Database connections (JDBC, ODBC)
- Almost all report elements with the exception of custom elements
- Business objects (dimensions, measures, attributes) and their definitions using pure SQL

Examples of Unsupported Features

> **Note**
> These are some examples of SAP BO features that MANTA doesn't support yet. Please note that this list is not complete. We plan on supporting most of the listed features in future releases.

- Data sources of data providers other than UNX universe (including UNV universes, files, etc.)
- Multisource-enabled data foundations
- @Functions (with the exception of @Aggregate_Aware, which is currently supported in some simple cases)
- Filter business objects
- Custom report elements

Qlik Sense

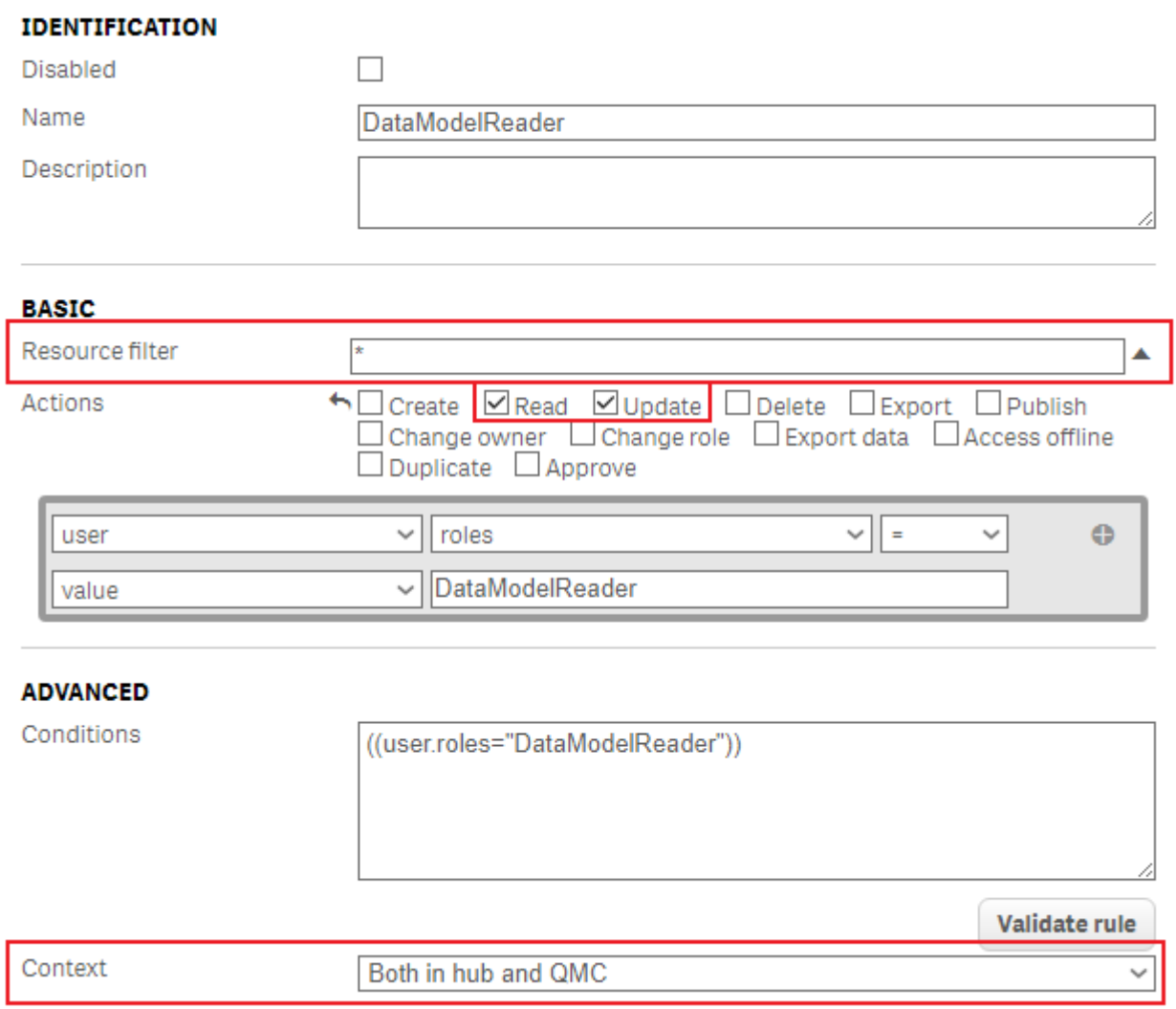
Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

The scanner has been developed for Qlik Sense September 2019 IR (13.42.1). Other versions of Qlik Sense may work, but we cannot guarantee it. (Qlik suggests always keeping Qlik Sense up to date.)

Prerequisites

- Running Qlik Sense Desktop or Qlik Sense Enterprise distribution (v13.42.1)
  - Qlik Sense Business is not supported because it is not possible to connect to this distribution remotely.
  - In the case of Qlik Sense Desktop, MANTA will be run on the same machine as the Qlik Sense Desktop distribution.
- Qlik Sense Enterprise—a user and their user directory with the following privileges
  - Access to the stream where the application that will be extracted is published. Without this, the application is not visible to the extractor.
  - The user + user directory combination needs the following security rule in their role (which can be assigned according to the *Security Rules* section of the QMC).

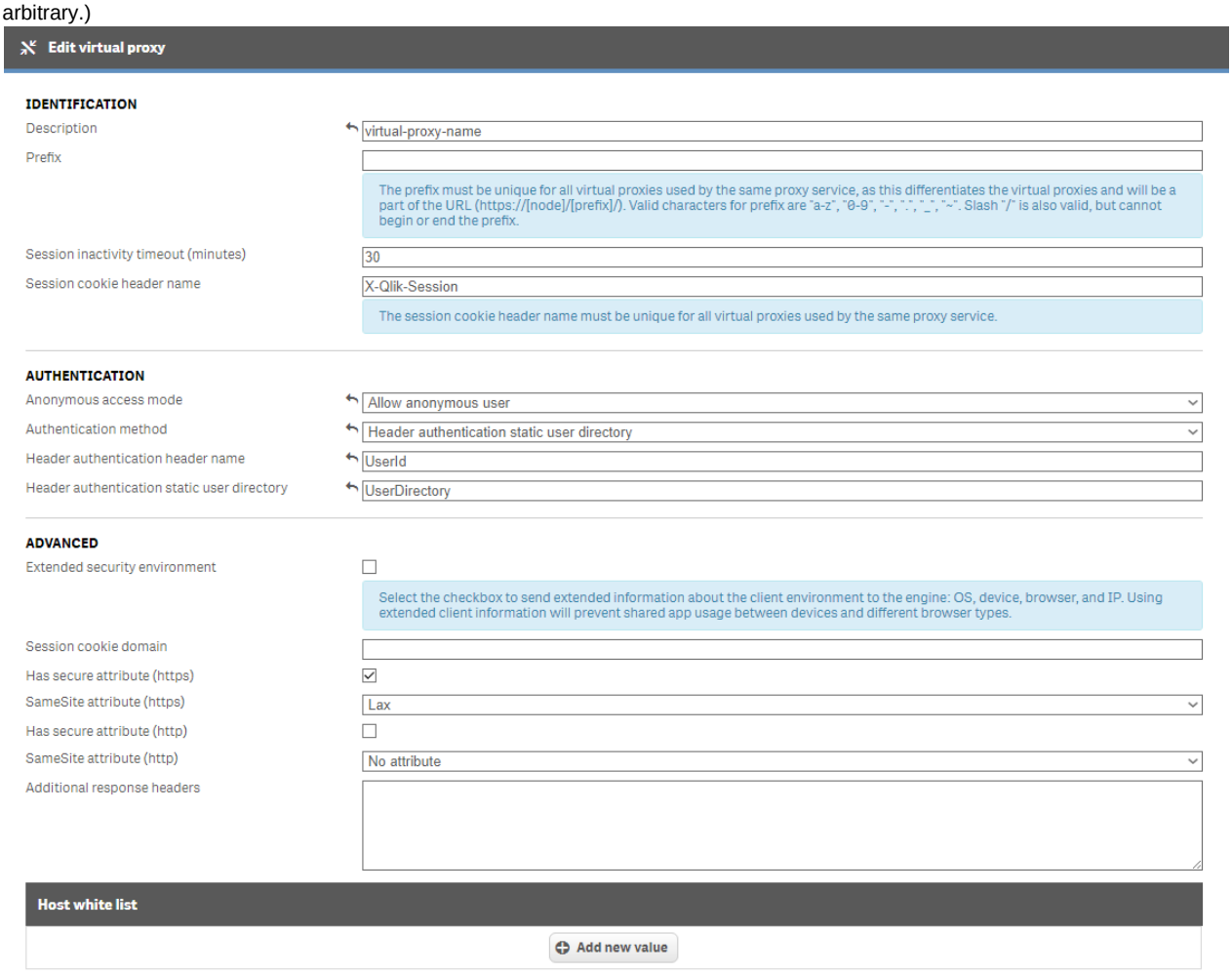


  - If the user + user directory does not have this security rule in their role, the *DataModel* cannot be extracted and some data flow may be missing.
- There are different sets of steps that need to be done in the Qlik Management Console (QMC; usually your Qlik Sense Enterprise server administrator performs those) and on the machine that runs MANTA. Expand the panels below to see instructions.

Steps to be done in the Qlik Management Console (typically by a Qlik Sense server administrator)

1. Go to the Qlik Sense Enterprise server's Qlik Management Console (QMC) and navigate to the *Certificates* section (URL: *your.server.domain/qmc/certificates*).

2. Set the *Certificate Password* (not empty; later referred to as *KeyPassword*) and input the extracting computer's *Machine Name.* (The machine name can be any text. It is simply used to create a folder of that name to store the certificates.)
3. For the *Export Format* field, select *Windows Format.*
4. Click on the button *Export Certificates.*
5. Go back to the server's QMC and navigate to the section called *Virtual Proxies.*
6. Create a new virtual proxy and fill in the fields as shown below. (Leave the fields not shown at their default values; the description is arbitrary.)



7. Add the IP address of the machine that will run MANTA to the virtual proxy's whitelist.
8. Save the changes.

Steps to be done on the machine running MANTA (usually by the MANTA user)

1. There will be three certificates in the export location: *client.pfx, root.cer,* and *server.pfx.* The export location can be found under the form on the *Certificates* page.
2. If you are running MANTA 1.32 or earlier, you have to import the root certificate (*root.cer*) to the client private keystore (*client.pfx*) manually. Ignore this step if you are running MANTA 1.33 or later.
   a. Import the *root.cer* certificate to the *client.pfx* keystore. There are two simple ways to do this:
      i. Use a keystore program with a graphical user interface such as *KeyStore Explorer*. If using *KeyStore Explorer:*
         1. Use *Open an Existing KeyStore* to open *client.pfx*.
         2. The utility will ask you for a keystore password. This is the password you set in step 2 above when exporting the certificates.
         3. Drag and Drop *root.cer* into the keystore and select *Import* (or select the import tool and select *root.cer.* Accept the default for the *Trusted Certificate Alias*.
         4. Click *OK* twice to exit the import.
         5. Save the KeyStore.
      ii. Use Java's *keytool* utility in the command line: `keytool -importcert -file root.pem -keystore client.pfx -alias qlikcert`.
         1. The alias `qlikcert` can be changed or omitted completely with the `-alias` flag.
         2. The utility will ask you for a keystore password. This is the password you set when exporting the certificates in step 2 above.
3. Remember the path where the keystore was created. You will need to specify it in the `qliksense.extractor.pathToKeystore` configuration property.

Please note that other security rules on your server may prevent the scanner from successfully extracting the data. If you experience problems with the extraction, please review your security rules, checking for those that may be problematic.

Qlik Sense Scanner Overview

The scanner uses Qlik Sense's JSON API to communicate with the server and extract all metadata necessary for data flow analysis. With each server request, it retrieves a piece of information about individual objects in a Qlik Sense application—about sheets, report items, tables used, or master items. Once all this data has been collected, the scanner crawls through all the server responses to put together all the pieces of the puzzle. Once the application with all its child objects is put together, you can analyze individual parts, find relations and data flows between objects, and create a dataflow graph. Parse and analyze the application's load script to accurately visualize how data is loaded and transformed before it is used in the Qlik Sense application. After that, you can analyze all expressions and table column uses across all sheets, report items, dimensions, measures, and their child objects to generate dataflow graphs. Once these graphs are created, adjust them to make them more readable and remove unimportant parts.

Supported Features

This is a list of Qlik Sense features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

Extraction

- A fully functioning extractor from a Qlik Sense Desktop server
- A fully functioning extractor from a Qlik Sense Enterprise server (depends on credentials and sufficient privileges)

Data Flow Analysis

- Parsing the load script
- Analyzing parts of the load script (see below)—visualizing how data is loaded and transformed before it is used in a Qlik Sense application
- Analyzing the "analysis" part of Qlik Sense—sheets, visualizations, and their child objects
  - Sheets
  - Master items—dimensions, measures, and visualizations (in the case of visualizations, only the visualization types listed below)
  - Visualizations and their properties—bar chart; box plot; combo chart; container; distribution plot; filter pane; gauge; histogram; KPI; line chart; map; pie chart; pivot table; scatter plot; table, text, and image; treemap; waterfall chart; (as of MANTA Flow R36) Mekko chart
- Visualization and load script expression analysis, including set expressions

Load Script Statement Analysis

There are some special cases where Qlik Sense accepts a form of a load script statement that is not described in the documentation or is declared invalid. In such cases, parsing may fail and some data flows may be missing.

- Reading Load Script files included with the *include* or *must_include* dollar expansion variants
- Variable value tracking (simple strings) and manual mapping for variable values
- The statements listed below use hard-coded field names
- LOAD
  - Loading from a local file from either a folder connection or the working directory (**not a web file**)
  - Loading from a previously-loaded field
  - Loading from fully-supported inline data (including a self-defined delimiter)
  - Loading from a previously-loaded table (RESIDENT keyword)
  - Loading from file and from table fields using asterisks in the field list
- Autogenerate LOAD
- Loading data from a succeeding table
- DROP field/table
- STORE
- SQL (no variable value replacement)
- RENAME field/table
- SELECT (no variable value replacement)
- DIRECTORY
- CONNECT (only in versions that define connections as `LIB CONNECT TO <connection-name>`)
- SUB and CALL (supported as of MANTA Flow R35)
- ALIAS (supported as of MANTA Flow R36)

Unsupported Features

Extraction

- Extraction from a Qlik Sense Business server

Data Flow Analysis

- Visualizations—button, auto-chart (charts created by Qlik Sense when the *Chart Suggestions* option is switched on), custom visualizations
- Latest visualization properties (KPI link to sheet, line chart coloring, tooltip, etc.)
- Measures trend lines
- Analysis of the “stories” in Qlik Sense
- Indirect data flows

Load Script Statement Analysis

- Direct query statement (parsing of this statement may fail; the statement structure is not completely defined)
- Prefixes of LOAD/SELECT/MAP statements are ignored
- No wildcards are being evaluated (asterisks or ?-like wildcards) in expressions except for the LOAD statement where the asterisk is analyzed
- LOAD statement:
  - Loading data where the fields are defined by their position
  - Loading data from an analytic connection
  - Analysis of WHERE/WHILE/GROUP BY/ORDER BY clauses
- Legacy scripting mode not supported
- CONNECT statement which defines connections in legacy mode
- Other statements are ignored (not analyzed)

SAS

The following are the prerequisites necessary to connect MANTA to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- SAS scripts with the `.sas` file extension to be available on a filesystem accessible to MANTA
- List of pre-assigned libraries along with their definitions
- Version 9.4 is officially supported. Other versions of SAS have not been tested but are expected to work. Please contact the help desk if you have any issues.

Supported Features

This is a list of SAS features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

- Top-level statements
  - Libname, Option, Run, Quit statements
    - SAS/ACCESS Interface to JDBC, ODBC, ORACLE, POSTGRESQL, NETEZZA, TERADATA, DB2 (as of R36), MS SQL (as of R36)
  - Filename (as of R38)
  - SAS/CONNECT
    - Rsubmit, Signon, Signoff, Rdisplay, Rget, Waitfor, Listtask, Killtask (as of R38)
    - Proc download (as of R38)
- Data step statements
  - Control flow—if, if-then-else, do-end, do-while, do-iterative
  - Statements
    - input
    - assign
    - length
    - attrib
    - set
    - merge
    - datalines
    - delete
    - put (as of R33)
    - keep (as of R33)
    - error (as of R33)
    - retain (as of R34)
    - rename (as of R34)
    - by (as of R34)
    - return (as of R36)
    - stop (as of R36)
    - abort (as of R36)

    - informat (as of R37)
    - infile (as of R38)
- Proc step statements
  - Proc SQL statements (as of R31)
    - alter table
    - create index
    - create table/view
    - select (without select directly from SAS file)
      - support of calculated columns (as of R37)
    - connect to
    - disconnect
    - execute
    - delete
    - describe
    - drop
    - insert
    - reset
    - update
    - validate
  - Proc fcmp (as of R31)
    - function
    - subroutine
  - Proc freq (as of R31)
    - by, exact, output, tables, test, weight
  - Proc append (as of R33)
  - Proc sort (as of R34)
  - Proc datasets
    - delete (as of R36)
    - copy (as of R38)
  - Proc contents
    - delete (as of R37)
- Macros
  - Macro variables (limited)
  - Macro methods (limited)

Data Modeling Tools

E/R Studio

The following are the prerequisites necessary to connect MANTA to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including their performance.

- Supported versions: ER/Studio 17 or newer
- Files with models in DM1 format

Erwin

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Erwin Data Modeler versions r9.8 to r12.0 are officially supported.
- Other versions of Erwin Data Modeler have not been tested but are expected to work. Please contact the help desk if you have any issues.
- Currently, the logical, physical, and logical/physical Erwin models can be imported to MANTA.
- The appropriate `*.xml` model files must be placed in the MANTA CLI input folder in the subfolder with the name matching the model system. Note that the binary `.erwin` file format is **not supported**. Models must be saved as `.xml` in order to be processed by MANTA.
- **Only** the XML standard format is supported. Note that the XML min and XML repository formats are **not supported**.

SAP PowerDesigner

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- PowerDesigner versions from 15 to 16.7 are officially supported.
- Other versions of PowerDesigner have not been tested but are expected to work. Please contact the help desk if you have any issues.

- Currently, the conceptual, logical, and physical PowerDesigner models can be imported to MANTA.
- The appropriate model files (`*.cdm`, `*.ldm`, and `*.pdm`, respectively) must be placed in the MANTA CLI input folder in the subfolder with the name matching the model system.

Programming Languages

Cobol/JCL

Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary to connect MANTA to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- ANSI/ISO COBOL 85 compatible scripts
  - File extension of scripts must be `.cbl` or `.cob`
  - Each script has one supported formatting: FIXED, HP TANDEM, VARIABLE, or FREE
- Copybooks (optional)
  - File extension of copybooks is irrelevant
- IBM Mainframe Job Control scripts, also known as JCL scripts (optional)
  - File extension of JCL scripts must be `.jcl` and the extension of includes must be `.in`

Please note that additional extensions used in Cobol scripts that were added in newer ANSI/ISO standards (object programming, intrinsic functions, etc.) and dialect (compiler) specific extensions (embedded SQL, etc.) may be supported but are not guaranteed.

Java

Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

- Java SE Runtime Environment 8 (JRE 8)

Although it is possible to run Java analysis using newer versions, JRE 8 libraries are still required and their location must be configured in the `javaCommon.properties` file when running on JRE 9 or newer.

Supported Java technologies:

- Java language—arrays, control flow, expressions, operations, inheritance, etc.
- Java SE
  - Standard Java collections in the `java.util` package
  - Standard JDBC connections in the `java.sql` package
  - Standard I/O operations in `java.io` packages
- Java EE
  - Java API for RESTful web services (JAX-RS)
  - Data sources based on the `javax.sql.DataSource` interface
  - Support for injection via `javax.inject.Inject` annotation
- Spring Framework
  - Dependency injection configured in XML and property files
  - Spring JDBC template
  - Spring MVC
- Kafka framework (producer and consumer)
- MyBatis framework
- Spark framework (v2.4)—Spark SQL
  - Dataset API
  - Reading and writing data using DataFrameReader and DataFrameWriter
    - CSV and JSON files; databases using JDBC methods
  - The following are not supported:
    - Raw RDDs
    - Configurations using external files (e.g., properties)
    - Raw SQL parsing (e.g., temporary views, persistent tables, the `Dataset::selectExpr` method)

The application being analyzed must be packed as follows (and include some kind of Control Flow Entry Point).

Distribution format	Description
Dedicated directories for application and library JAR files	Application JAR files placed in the defined directory, which must not contain library JARs or classes; any libraries used in the Java code must be provided as JAR files and placed in the defined directory
Single directory with standard JAR files	A mix of application and third-party JAR files within a single directory
Standard WAR file	A standard specialization of a JAR file with a well-specified structure that is a container for server applications
Uber-JAR file (also known as a fat JAR)	A JAR file that combines application and library code with resources as a result of unpacking all the original application and library JAR files and repackaging the extracted content into one (technically standard) JAR file ⚠ Please note that uber-JAR files are problematic on their own. Some information is lost during creation such as: › It's (generally) not possible to determine which original JAR file a particular file in the resulting uber-JAR came from. Although it's possible to guess for `*.class` files, for all other (e.g., configuration) files it's impossible. › Any duplicate file names among all source files need to be handled somehow. Typically, either the first or last one found wins (and the rest are simply discarded) or they are specially handled (e.g., by merging their content). For these reasons, it's not even possible for the Bytecode Extractor to extract uber-JAR files correctly, and further user intervention may be required to customize/correct the resulting output in order for the following dataflow generation to succeed. The recommended solution is to use a more capable distribution format that doesn't have such limitations such as a Spring Boot executable JAR file (which is very easy to produce using a Maven or Gradle plugin).
Spring Boot executable JAR file	All application code packed together with the libraries in one JAR file containing other JAR `files/*.class` files in a well-defined format
Spring Boot executable WAR file	All server application code packaged together with the libraries (including those normally provided by a servlet container / application server) in one WAR file containing other JAR `files/*.class` files in a well-defined format

C#

Note
Please note that this scanner is considered experimental and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance. C# Scanner does not require any installations other than the MANTA Flow application.

Supported C# libraries and frameworks:

› ADO.NET with MS SQL connections
› Console I/O dataflows—System.Console
› Filestreams I/O—System.IO.TextReader, System.IO.TextWriter, System.IO.FileStream, System.IO.File

Unsupported C# language features:

› Lambda methods and delegates
› Multithreading and parallelism
› Reflection and dynamic code generation
› Dynamic language runtime

The application being analyzed may be provided as:

› `.dll` or `.exe` files
› C# project directory with a `.csproj` file and C# source code (project compatible with .NET Core 3.1)

- `.cs` files with C# source code (without non-standard dependencies and compatible with .NET Core 3.1)

Published applications with native code (for example, single-file with runtime) are currently not supported.

Python

⚠ Please note that this scanner is considered experimental, and MANTA recommends using it for testing purposes only.

The following are the prerequisites necessary for MANTA to connect to this third-party system, which you may choose to do at your sole discretion. Please note that while these are usually sufficient to connect to this third-party system, we cannot guarantee that the connection or integration will be successful since we have no control, liability, or responsibility for third-party products or services, including for their performance.

Prerequisites

- MANTA Flow R35 or newer:
  - Installation of a Python 3.x interpreter
  - Installation of SQLAlchemy

Supported Features

ℹ This is a list of Python features that MANTA supports. There may be features that aren't explicitly named but are included in the named items, as this list aims to be a high-level overview. Otherwise, features that are not listed are primarily considered not supported.

Extraction

- Extraction of Python programs from ZIP archives
- Extraction of Python programs from directories on the local filesystem
- Extraction of the builtins library from the Python interpreter
- Extraction of the database libraries SQLAlchemy
- MANTA Flow R37 or newer:
  - Extraction of the AWS SDK library boto3
  - Extraction of PySpark libraries
  - Extraction of Pandas libraries

Data Flow Analysis

- Import resolution (if the imported modules are extracted and analyzed as well)
- Dataflow analysis of expressions, assignments, function invocations, and other standard components of the language
- Python builtins analysis
- File system dataflow generation for the standard library functions `print`, `input`, `read`, and `write`
- Database dataflow generation for the SQLAlchemy functions `create_engine`, `connect`, `execute`, and `all`
- File system dataflow generation for the CSV and JSON library functions `json.load`, `json.loads`, `json.dump`, `json.dumps`, `csv.reader`, `csv.writer`, `csv.writerow`, and `csv.writerows`
- Support for String operations
  - String concatenation—`string = 'Man' + 'ta'`
  - String join—`','.join(['column_a', 'column_b', 'column_c'])`
  - String format—`"SELECT {}, {}, {} FROM {}".format(column_a, column_b, column_c, table_name)`
    - Covers all syntax defined by PEP 3103 and PEP 498
    - Positional formatting is not supported: `"SELECT {2}, {1}, {0} FROM {3}".format(column_c, column_b, column_a, table_name)`
  - MANTA Flow R37 or newer:
    - Python f-strings—`f"formatted string in Python {version}"`
- Precise dataflow analysis of the named or indexed column
  - MANTA Flow R37 or newer:
    - For databases
  - MANTA Flow R38 or newer:
    - For local CSV files
- MANTA Flow R37 or newer:
  - Support for additional libraries and frameworks
    - PySpark
    - Pandas
- MANTA Flow R38 or newer:

- Support for additional libraries and frameworks
  - AWS SDK—boto3

Unsupported Features

- Extraction of installed external libraries (unless stated otherwise)
- Analysis of entry-point data flows
- Analysis of Python 2.x programs

Other

Filesystem

- File Path Mapping Configuration
- Mapping Rules
- Source Technology Values
- Examples

A data file is represented in a dataflow graph as a `File` node inside an appropriate `Directory` node hierarchy and, finally, inside a `Server` node.
For example: `C:/data/file.txt` will be represented as `localhost[Server] C: [Directory] data [Directory] file.txt [File]`.

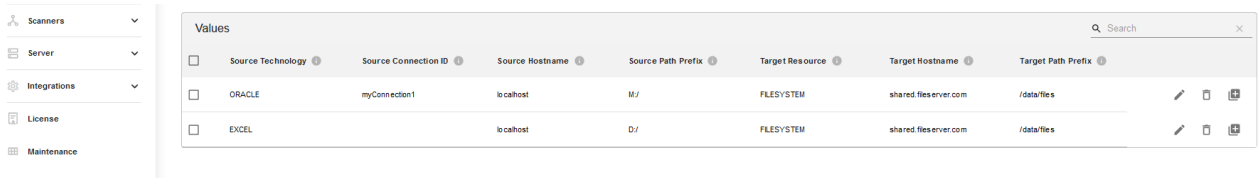
File Path Mapping Configuration

This configuration allows the mapping of filesystem paths to provide a more user-friendly representation and/or to ensure that the same file can be correctly referenced from different technologies, even if the technology itself references the file by a different file path (for example, if the hostnames are different due to the network configuration). The configuration is common to all source systems.

The mapping is configured in a standard CSV file with the following columns.

Column	Description	Examples / allowed values
Source Technology	The source technology (source system type) this entry applies to. (See the complete list of technologies below.) If empty, the mapping entry may be applied to any technology.	ORACLE DATASTAGE …
Source Connection ID	The connection ID of the connection this entry applies to (case sensitive). If empty, the mapping entry may be applied to any connection.	oracle-uat
Source Hostname	Hostname this entry applies to (case-insensitive). If empty, the mapping entry may be applied to paths with any hostname.	localhost
Source Path Prefix	Source path prefix (regular expression) to be matched. If this is non-empty, the entry may only be applied to file paths that start with the given prefix (disregarding the hostname). The matched path prefix may then be replaced by a different prefix during the mapping.	C:/data
Target Resource	Graph resource to use for the resulting graph nodes. If empty, the original resource provided by the source system scanner will be used.	FILESYSTEM AMAZON_S3
Target Hostname	Hostname to use for the resulting graph representation of the file. If empty, the original hostname provided by the source system scanner will be used.	fileserver.int.example.com
Target Path Prefix	Path prefix to use for the resulting graph representation of the file. If this is non-empty, any path prefix matched by the source path prefix expression will be removed and the given target path prefix will be prepended to the remaining file path.	data/dwh/stage

The configuration can be changed in *Configuration* > *CLI* > *Common* > *File Path Mapping* in MANTA Admin UI, as seen in the screenshot below.



Mapping Rules

- The first matching record is used
- Any condition that is left empty in the configuration is always met
- The record matches if all four source conditions are met:
  - Source Technology
    - Identification of the technology that referenced the file (e.g., `ORACLE`, `DATASTAGE`—see below)
  - Source Connection ID
    - Identification of the connection where the file is used (the same as that defined when creating the connection)
    - Case sensitive
  - Source Hostname
    - Hostname of the file as used in the analyzed technology
    - Case insensitive
  - Source Path Prefix
    - Beginning of the file path to be matched and replaced (when mapping file paths)
    - Regular expression
    - Cannot be used only to remove a prefix, but can be used only to add a prefix
    - Case sensitivity is controlled by the `filepath.lowercase` property
- Non-empty target values from the matched record are used to override the original attributes
  - Target Resource
  - Target Hostname
  - Target Path Prefix
    - Replaces the section of the path matched by the source path prefix

Source Technology Values

Value to use	Technology
ANNOTATED_SCRIPT	Open MANTA Annotated Script Scanner
BIGQUERY	Google BigQuery
BYTECODE	Bytecode
COBOL	Cobol
COGNOS	IBM Cognos Analytics, Cognos Business Intelligence
CONCEPTUAL_OVERLAYS	Open MANTA Conceptual Overlays
CSHARP	C#
DATASTAGE	IBM InfoSphere DataStage
DATAFACTORY	Azure Data Factory
DB2	IBM DB2
DIRECT_LINKS	Open MANTA Direct Links
ERSTUDIO	Idera ER/Studio Data Modeling Tools
ERWIN	Erwin Data Modeler
EXCEL	Microsoft/Azure Excel
EXTENSIONS	Open MANTA Extensions
FIVETRAN	Fivetran
HIVE	Apache Hive
IFPC	Informatica PowerCenter

INTERPOLATION	Open MANTA Conceptual Overlays Lineage Interpolation
JAVA	Java
MSSQL	Microsoft SQL Server, Parallel Data Warehouse (PDW), Analytics Platform System (APS), Azure SQL Database, Azure SQL, Amazon RDS for SQL Server
NETEZZA	IBM Netezza (IBM PureData System for Analytics)
OBIEE	Oracle Business Intelligence Enterprise Edition
ODI	Oracle Data Integrator
ORACLE	Oracle Database, Exadata
PIGLATIN	Pig Latin
POSTGRESQL	PostgreSQL, Greenplum, Amazon Redshift, Yellowbrick, Amazon RDS, or Amazon Aurora for PostgreSQL
POWERBI	Microsoft/Azure PowerBI
POWERDESIGNER	SAP PowerDesigner
PYTHON	Python
QLIKSENSE	Qlik Sense
SAPBO	SAP Business Objects
SAS	Statistical Analysis Software
SNOWFLAKE	Snowflake
SQOOP	Apache Sqoop
SSAS	Microsoft/Azure SQL Server Analytical Services
SSIS	Microsoft/Azure SQL Server Integration Services
SSRS	Microsoft/Azure SQL Server Reporting Services
STREAMSETS	StreamSets
TABLEAU	Tableau
TALEND	Talend ETL
TERADATA	Teradata Database, BTEQ, Teradata Parallel Transporter (TPT)

Examples

Scenario			Filesystem resource mapping			
Description	**Original path**	**Desired path**	**Source hostname**	**Source path prefix**	**Target hostname**	**Target path prefix**
Move and rename **directory** (data) to a directory of the known Filesystem (BillingServer), keeping child directories the same	/Filesystem/localhost/data/files/	/Filesystem/BillingServer/Audit/files/	localhost	/data	BillingServer	/Audit
Move and rename **file** to a directory of the known Filesystem (BillingServer)	/Filesystem/189.89.89.89/az_38x.txt	/Filesystem/BillingServer/Address/Arizona.txt	189.89.89.89	az_38x.txt *(NO SLASH)*	BillingServer	/Address/Arizona.txt
Hide a **drive** in the path name	/Filesystem/external_1/c:/users/sales/	/Filesystem/external_1/users/sales		c: *(NO SLASH)*		
Rename a **file** consisting of REGEX special characters	/Filesystem/tmp/parameters/#param	/Filesystem/tmp/parameters/contract_address.dat		#paramset.\$src_dir_parm.contract_address.dat *('\' to escape '$')*		contract_address.dat

	set.$src_dir_parm. contract_address. dat					

MANTA Server Administration

HTTPS Configuration

To enable HTTPS communication, please follow these steps.

1. Create a keystore with a certificate.
   - Apache Tomcat needs a certificate to identify itself. You can create a self-signed certificate or import a certificate signed by your CA.
   - To create a self-signed certificate:
     - Windows: `"%JAVA_HOME%\bin\keytool" -genkey -alias tomcat -keyalg RSA -keystore \path\to\my\keystore`
     - Unix: `$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore /path/to/my/keystore`
   - You can find more details on the Tomcat homepage.
   - If you have a private key that is already available for use, follow these steps to import it to a new keystore (Linux).
     - `openssl pkcs12 -export -in <base>.cer -inkey <private>.key -out server.p12 -name tomcat -CAfile <parent_cert>.cer`
     - `keytool -importkeystore \ -deststorepass [changeit] -destkeypass [changeit] -destkeystore server.keystore \ -srckeystore server.p12 -srcstoretype PKCS12 -srcstorepass [some_password] \ -alias tomcat`
     - `keytool -import -trustcacerts -file <root_cert>.cer -alias tomcatroot -keystore server.keystore`

The private key password may be different from the keystore password. When this scenario occurs, MANTA Server will not be able to locate the SSL certificate and will throw an error message in the `server/logs/catalina.out` file. To resolve this issue, reset the private key password to match the keystore password and restart MANTA Services.

1. Both Admin UI and MANTA Server are regular web applications up to R37; as of R37 Admin UI has been changed to SpringBoot and because of that the configuration is different. Please apply the steps according to your version of MANTA.
   a. **For Flow Server (all versions) and Admin UI prior to R37** modify the `<MANTA_SERVER_HOME>/conf/server.xml or <MANTA_ADMIN_HOME>/conf/server.xml` file.
      i. Add the *Connector* configuration for HTTPS to the `<Service>` tag. Note: It is possible to choose a port other than 8443 for HTTPS, but ports 80 and 8080 should be reserved for HTTP connectors, even if they are disabled. We therefore recommend using port 443 or 8443, if available.

In Linux, ports under 1024 are considered reserved for the operating system. A non-root user will not be able to configure port 443 on Linux for this reason. In this scenario, choose any available port above 1023 for your HTTPS communications with MANTA.

server.xml

```
<Server ...>
    <Service ...>
        <Connector port="8080" ... /> <!-- HTTP Connector will
be already present in the configuration -->

        <!-- HTTPS Connector is added below -->
        <Connector
            protocol="org.apache.coyote.http11.
Http11NioProtocol"
            port="8443" maxThreads="200"
            scheme="https" secure="true" SSLEnabled="true"
            relaxedQueryChars="[ ]" >
                        <SSLHostConfig>
                <Certificate certificateKeystoreFile="${user.
```

```
home}/.keystore"
                              certificateKeystorePassword="
changeit"
                              type="RSA"
                              sslProtocol="TLS" />
            </SSLHostConfig>
        </Connector>
...
```

ii. Change the attributes `certificateKeystoreFile` and `certificateKeystorePassword` regarding the keystore created in step one.
iii. If the private key that should be used for connection encryption is stored under a different alias than *Tomcat* in the keystore, add the attribute `keyAlias="actual_key_alias"` to the *Certificate* element above.
iv. In this section, you can modify the port for HTTPS. It is configured in the attribute `port`. If the port is changed, it is also important to change it in the section for standard HTTP communication to `redirectPort`. (Optional)
v. The configuration described will use NIO implementation. Optionally, you can change the implementation to:
   1. Legacy implementation by changing the *Protocol* attribute to "`org.apache.coyote.http11.Http11Protocol`" (not recommended)
   2. NIO2 implementation by changing the *Protocol* attribute to "`org.apache.coyote.http11.Http11Nio2Protocol`" (not recommended)
   3. APR implementation by following the official Tomcat guide

b. **For Admin UI from as of R37** modify the `<MANTA_ADMIN_HOME>/WEB-INF/conf/application.properties` file.
   i. Add properties for HTTPS.
   Note: It is possible to choose a port other than 8443 for HTTPS, but ports 80 and 8080 should be reserved for HTTP connectors, even if they are disabled. We therefore recommend using port 443 or 8443, if available. Instructions for changing the port are here

> In Linux, ports under 1024 are considered reserved for the operating system. A non-root user will not be able to configure port 443 on Linux for this reason. In this scenario, choose any available port above 1023 for your https communications with MANTA.

application.properties

```
...
manta.admin-ui.ssl.enable=true
manta.admin-ui.ssl.key-password=changeit
manta.admin-ui.ssl.key-store=${user.home}/.keystore
manta.admin-ui.ssl.key-store-type=pkcs12
manta.admin-ui.ssl.key-store-password=changeit
manta.admin-ui.ssl.key-alias=
...
```

   ii. Change the properties **key-password**, **key-store**, **key-store-type** and **key-store-password** regarding the keystore created in step one.
   iii. If the keystore contains more than one key, it is necessary to fill the alias for the key into the **key-alias** variable.

2. Change the URL appropriately in any client that communicates with the server that should utilize HTTPS.
3. If HTTPS enforcement is enabled on the server, the MANTA Service Utility will need to be updated to communicate with the server via the HTTPS protocol as well. To do this:
   a. Log in to Admin UI and navigate to Configuration > CLI > Common > Common Config.
   b. Update the Repository Server URL property to utilize the HTTPS protocol as well as the newly configured port number.
   c. Use the MANTA System CLI Truststore *Settings* button to load MANTA Server's public certificate to the `mantaSystemTruststore.pkcs12` truststore.
      i. You may need to export this from MANTA Server's keystore. You can complete this by running the following command.
      `keytool -export -keystore [examplestore] -alias [alias_name] -file [Example.cer]`
4. Once the steps above have been completed, save the changes, restart the modified service, and test the connection by executing the *Diagnose Repository Scenario* from Process Manager.

SSO Configuration

Encoding

All property files are encoded in ISO 8859-1 character encoding. Characters that cannot be directly represented in this encoding can be written using Unicode escapes as defined in section 3.3 of The Java™ Language Specification; only a single "u" character is allowed in an escape sequence.

Authorization

This article describes the overall concepts of authentication and authorization in MANTA Flow Server. There are two guides that explain how to configure the authentication. Choose the right guide according to the version of MANTA you are using.

- For R34 and older, see Legacy Authentication (R34 and Older).
- For R35 and newer, see MANTA Identity Management: Keycloak.

User Roles

The application uses several user roles to authorize specific operations within the metadata repository. The roles are:

- ROLE_USER—basic role needed for all secured pages, which by default is all pages
- ROLE_EXPORTER—exports data from the repository
- ROLE_MERGER—executes data-modifying operations such as merging objects, truncating databases, and propagating edges
- ROLE_VIEWER_DATAFLOW—executes operations for the visualization of data flow
- ROLE_VIEWER_CATALOG—explores and searches the metadata catalog
- ROLE_USAGE—exports and cleans server-usage metadata
- ROLE_REPOSITORY_READ—executes data-reading operations via repository API
- ROLE_REPOSITORY_WRITE—executes data-modifying operations via repository API
- ROLE_REPOSITORY_EVALUATE—executes special data-modifying operations based on DSL-script evaluation via repository API
- Additional roles can be created simply by starting to use them as described below; creating new roles is only useful for the section called "Access Rights for the Metadata Repository"

Disabling Security for Specific Parts

⚠ Please note that as of MANTA R35 this change will not be persisted when upgrading to newer versions of MANTA.

This configuration is no longer available as of R38.

It is possible to disable the authorization and authentication for specific parts of the MANTA Flow Server. For example, it is possible to view data flows without logging in.

The security definition is in the file `<MANTA_SERVER_HOME>/server/webapps/manta-dataflow-server/WEB-INF/classes/securityContext.xml`.(The configuration file is located in the `classes` folder, not the `config` folder, prior to R35.) There you will find a tag `sec:http` that contains access to the rules. Two lines of the configuration have to be added to allow anonymous access to some parts of the application.

- The URL pattern which should be permitted
  - `<sec:intercept-url pattern="URL_PATTERN" access="permitAll"/>`
  - For the dataflow visualization, the `URL_PATTERN` would be `/viewer/**`
- Roles which are granted to the anonymous user
  - `<sec:anonymous granted-authority="ROLE1,ROLE2"/>`

Access Rights for the Metadata Repository

It is possible to define access rights for the metadata repository. This means that some parts of the metadata repository may only be visible to particular users.

Enabling and Disabling Access Rights

To enable this feature, set the `repository.permissions-enabled` property in the configuration *Configurations* > *Server* > *Common* > *Repository Configuration* in MANTA Admin UI to true.

To disable this feature, set that property to `false`. In this case, the entire repository will be accessible to all users.

Defining Access Rights

If the access rights feature is enabled, it is possible to configure metadata repository permissions for MANTA users. This is done on three levels.

Level	Location of definition	Description

Repository view configuration	MANTA	A repository view is any part of the repository. It is defined as a set of included and excluded repository subtrees.
Assigning repository views to user roles	MANTA	Each role has a set of repository views assigned to it that are accessible to users with this role. It is possible to assign no view (i.e., no part of the repository is accessible) as well as the entire repository. One view can be assigned to more than one role.
Assigning users to user roles	External systems (e.g., LDAP) or MANTA	Each user can be assigned one or more roles and vice versa. There is no need to define the roles explicitly in MANTA.

Repository View Configuration

Repository views can be configured in the configuration *Configurations* > *Server* > *Common* > *Repository Views* in MANTA Admin UI. The rows are records of the inclusion of repository objects in the view or the exclusion of them from the view. The record fields are:

Repository view	Name of the repository view the record applies to
Type	Type of record. The value must be either `INCLUDE` or `EXCLUDE`. If it is `INCLUDE`, the affected objects are included in the view. If it is `EXCLUDE`, the affected objects are excluded from the view. The exclusions precede the inclusions. If only `EXCLUDE` records are defined for the view, the rest of the repository is considered to be included in the view.
Affected objects	Case-insensitive regular expressions of repository path entries separated by slashes (`/`). A repository object is affected if and only if: 1. each entry of its path matches the corresponding regular expression, 2. or any of its ancestors fulfill point one. The object resource is the first object path entry. Special cases: › To enclose a path entry in double quotes ("), use the `\"` sequence. › To use double quotes as part of the path entry, use the `\"\"` sequence. › To use a backslash (\) as part of the path entry, use the `\\` sequence. Note that a backslash in an object's path needs to be escaped twice—once for the CSV file and another time for the regex.

Assigning Repository Views to User Roles

The assignment of repository views to user roles can be configured in the configuration *Configurations* > *Server* > *Common* > *Repository Views Permissions* in MANTA Admin UI. The rows are records of view-to-role assignments. The record fields are:

Role	The user role the record applies to; must be unique within the file
Repository views	Comma-separated list of views accessible to the role; if set to `*`, users with this role have access to the entire repository

Applying the Changes

To apply changes to the CSV configuration files above, it is necessary to restart the MANTA Server or enter an HTTP GET request using the following format.

`http://<server_name>:<port>/manta-dataflow-server/api/refresh`

where the `<server_name>` and `<port>` are provided by your application administrator.

If the `repository.permissions-enabled` property has been changed, a MANTA Server restart is always necessary.

Repository Object Permission Evaluation

A repository object is accessible to the user if and only if it is contained in at least one repository view assigned to at least one of the user roles.

Example Configuration

repositoryViews.csv

```
"Repository View";"Type";"Affected Objects"
Teradata;INCLUDE;Teradata
OracleDwhExclParty;INCLUDE;Oracle/ORCL/DWH
OracleDwhExclParty;EXCLUDE;Oracle/ORCL/DWH/PARTY.*
MSSQLInstance;EXCLUDE;MSSQL/Server\\\\Instance/.*
```

repositoryViewsPermission.csv

```
"Role";"Repository Views"
ROLE_SYSTEM;*
ROLE_USER;Teradata,OracleDwhExclParty
```

Users with the `ROLE_SYSTEM` role have access to the entire repository.

Users with the `ROLE_USER` role have access to all Teradata databases and `ORCL.DWH` Oracle schemas, excluding all objects (tables, views, etc.) having a name starting with `PARTY`.

Repository

Back Up and Restore

Physical Copy

To back up the metadata repository:

- Ensure that no application is using the MANTA Flow Server.
- Stop the MANTA Flow Server application by either running the `bin/shutdown.(bat|sh)` script or stopping its installed service.
- Locate the database directory.
  - `webapps/manta-dataflow-server/WEB-INF/data/neo4j` (Neo4j database as of R36)
  - or `webapps/manta-dataflow-server/WEB-INF/db` (Titan database up to and including R35)
- Copy the database directory to a backup location.
- Start the MANTA Flow Server again by either running `bin/startup.(bat|sh)` or starting its installed service.
- Ensure that the MANTA Flow Viewer application is running on its configured URL.

To restore a previous backup of a metadata repository:

- Ensure that no application is using the MANTA Flow Server.
- Stop the MANTA Flow Server application by either running the `bin/shutdown.(bat|sh)` script or stopping its installed service.
- Locate the database directory.
  - `webapps/manta-dataflow-server/WEB-INF/data/neo4j` (Neo4j database as of R36)
  - or `webapps/manta-dataflow-server/WEB-INF/db` (Titan database up to and including R35)
- Remove the database directory.
- Copy the database directory from your backup location back to the original directory.
- Start the MANTA Flow Server again by either running `bin/startup.(bat|sh)` or starting its installed service.
- Ensure that the MANTA Flow Viewer application is running on its configured URL.

Physically copying the repository usually goes fast, but it depends on the OS and the version of MANTA Flow.

Dump

The metadata repository can be exported and imported in binary format, which is independent of the OS and the version of MANTA Flow. This dump applies to the whole repository including all versions and all technical metadata.

Dumps are only accessible to users with the role `ROLE_MERGER`.

Warning: Importing a dump will erase the old data in the target repository!

Links for the dump:

- Export – `<manta-server-url>/dump/export`
-

Import – `<manta-server-url>/dump/import`

A warning message appears when exporting a repository greater than 4 GB. The repository size threshold can be configured before the server starts using the `minSizeForExportWarning` property inside the `webapps/manta-dataflow-server/WEB-INF/classes/dump-context.xml` file.

Please note that as of MANTA R35 this change will not be persisted when upgrading to newer versions of MANTA.

REST API

It is also possible to export and import dumps through the REST API.

- Export—`<manta-server-url>/api/dump/export`
  - Method: `GET`
  - The dump is returned in response as `Content-Disposition: attachment`
- Import—`<manta-server-url>/api/dump/import`
  - Method: `POST`
  - `Content-Type: multipart/form-data`
  - The dump is sent in an attachment named "`file`"
  - Returns JSON with "isOK" flag and an error message upon failure

Maintenance Links

There are several links to maintain the repository.

- `<manta-server-url>/api/truncate`—permanently erases all data in the repository so that it is impossible to restore without a backup
- `<manta-server-url>/api/inspect`—collects information about the last revision made and displays the numbers for each type
- `<manta-server-url>/api/revision/rollback-current-revision`—rolls back the newest revision, which must be uncommitted
- `<manta-server-url>/api/revision/delete-last`—deletes the last committed revision (fails if there is an uncommitted revision)

Location of Persistent Files

The configuration `Configurations > Server > Common > Repository Configuration` in MANTA Admin UI contains the locations of all repository files and directories that should persist across MANTA Flow Server reboots.

Property name	Description	Examples
repository.storage.neo4j	The relative or absolute path to the directory with the Neo4j metadata database (as of R36)	WEB-INF/data/neo4j c:/manta/data/neo4j
repository.storage.dir	The relative or absolute path to the directory with the Titan metadata database (up to and including R35)	WEB-INF/db c:/manta/db
repository.sourcecode.dir	The relative or absolute path to the directory containing source codes	WEB-INF/sourcecode c:/manta/sourcecode
repository.usage-stats.db	The relative or absolute path to the database containing usage statistics; should not contain the database file extension	WEB-INF/usageStatsDb c:/manta/usageStatsDb
repository.script-metadata.db	The relative or absolute path to the database containing script metadata; should not contain the database file extension	WEB-INF/scriptMetadataDb c:/manta/scriptMetadataDb
repository.permissions.views	The relative or absolute path to the file containing the configuration of views of the metadata	WEB-INF/conf/repositoryViews.csv c:/manta/repositoryViews.csv
repository.permissions.view-permissions	The relative or absolute path to the file containing the permission configuration of views to the metadata	WEB-INF/conf /repositoryViewsPermission.csv c:/manta /repositoryViewsPermission.csv
repository.horizontal-filters	The relative or absolute path to the file containing the configuration of horizontal filters	WEB-INF/conf/horizontalFilters.json c:/manta/horizontalFilters.json
repository.horizontal-filter-groups	The relative or absolute path to the file containing the configuration of horizontal filter groups	WEB-INF/conf /horizontalFilterGroups.json

		c:/manta/horizontalFilterGroups.json

Advanced Configuration (Titan Database up to R35)

The advanced repository configuration is also available on the MANTA Admin UI page `Configurations > Server > Common > Repository Configuration`.

Property name	Description	Default value
storage.backend	Storage backend to be used for persistence (full class name of the StorageManager implementation or one of the pre-defined storage backends)	persistit
storage.buffercount	Size of the Persistit internal buffer	5000
storage.transactions	Enables transactional isolation in a storage backend	false
storage.maximum-pages	A property from the underlying Persistit database that determines how big the database can be—if the number of pages is greater, the graph database crashes and has to be cleaned so everything works again The property must be set before the MANTA repository is created; it does not have any effect afterwards	1000000000
cache.db-cache	Enables the database level cache	true
cache.db-cache-size	The size of the database level cache (percentage of the total heap space available to the Titan JVM when < 1.0, absolute when > 1.0)	0.3
cache.db-cache-time	Expiration time in milliseconds for elements held in the database level cache	10000
index.node-name	This property enables/disables fulltext index used for search features in MANTA Flow Visualization and is turned on by default. It only makes sense to turn it off when the MANTA instance is only used for integration with third-party tools and the visualization is not (ever) accessed by users. Then it may improve the performance of the MANTA scan. The property must be set before the MANTA repository is created. It does not have any effect afterward.	true

We strongly recommend that you do not modify anything without first discussing it with the MANTA Support Team.

Session Policy

It is also possible to adjust the behavior of MANTA Viewer sessions. The file that contains this configuration is `<MANTA_SERVER_HOME>/webapps/manta-dataflow-server/WEB-INF/web.xml`. The default configuration appears as follows.

```
<session-config>
    <cookie-config>
        <http-only>true</http-only>
    </cookie-config>
</session-config>
```

It is possible to change the default session timeout by adding the following entry in *session-config*. The number inside the tag is the timeout in minutes.

```
<session-timeout>30</session-timeout>
```

Cookies

By default, whenever a new session is created, the server generates a cookie as well as the JSESSIONID on the URL. When the server verifies that cookies are allowed, the JSESSIONID isn't necessary and is dropped. If the client comes back with no cookie, then the server needs to continue to use JSESSIONID rewriting in URL.

This behavior can be adjusted by adding the following entry in *session-config*. If the tracking mode is set to *COOKIE*, the JSESSIONID is not generated at all. (MANTA Server has to be restarted so that the configuration is applied.) Please note that if this configuration is active, MANTA Viewer will not work with disabled cookies.

```
<tracking-mode>COOKIE</tracking-mode>
```

Version Control System

The metadata repository supports the version control system. This system can theoretically hold an unlimited number of revisions, but each revision takes up space on the disk. The maximum number of revisions is maintained by the prune feature, which deletes older revisions that exceed the configured amount.

It is possible to turn off the version control system. One reason for this could be to only use the MANTA Flow Server for exporting to third-party applications. In this case, the client workflow must be changed.

Revision System

Two types of revisions exist.

1. Major revisions—work like revisions in older versions of MANTA. Each major revision is identified by an integer.
2. Minor revisions—enable the incremental update of repository metadata. Loaded metadata is merged with metadata from the previous revision. Each minor revision is identified by a six-digit float number.

Basic API

- POST `<manta-server-url>/api/revision/createMajor`—creates a new major revision
- POST `<manta-server-url>/api/revision/createMinor`—creates a new minor revision
- GET `<manta-server-url>/api/revision/commit/<revision>`—commits the revision `<revision>`
- POST `<manta-server-url>/api/revision/prune-preserve`—prunes old revisions; request parameters:
  - `revisionCount`—count of revisions to be preserved
  - `majorsOnly`—if `true` (by default), only preserved revisions are counted over major revisions; if `false`, all revisions are counted

Logging

The MANTA Dataflow Server uses the log4j2 framework for logging. The configuration file is placed in `<MANTA_SERVER_HOME>/webapps/manta-dataflow-server/WEB-INF/classes/log4j2.xml`. The official documentation for this framework can be found here.

Usage Statistics

The MANTA Flow Server collects usage statistics in the inner database. These statistics can be exported using the link `<manta-server-url>/usage` as an archived CSV file. The export form enables you to set the starting date from which the statistics are exported. The statistics can be cleaned via the URL `<manta-server-url>/usage/clean`.

The usage statistics CSV file has the following columns.

- *date* —The date and time of the action that is logged
- *user* —The user who performed/triggered the action
- *action* —The code of the action that was performed; the table below contains a complete list of action codes together with descriptions of the actions
- *params* —Additional parameters that are logged; the attributes are represented as a JSON object and are different for each action. The key in the JSON object is always the name of the parameter, and the value is either a value literal or a nested JSON object.

Action source module	Action code	Action description	Params

MANTA Flow	`server_start`	This action is logged after the start-up of the MANTA Flow Server application.	• *system*
			  – *osName*—OS name
			  – *javaVersion*—Java Virtual Machine implementation version
			  – *javaVnName*—Java Virtual Machine implementation name
			  – javaRuntimeVersion—Java Virtual Machine implementation version including build ID
			• *mainConfig—the* content of the `<MANTA_SERVER>/webapps/manta-dataflow-server/WEB-INF/classes/repository.properties` configuration file

MANTA Flow Viewer	`flow_catalog_show`	This action is logged when the user views the main screen of the MANTA Flow Viewer application.	N/A
MANTA Flow Viewer	`flow_vis_refView`	This action is logged when the user executes the analysis and visualization of the data lineage in the MANTA Flow Viewer application.	› *refViewTime*—the time needed to calculate the data flow in ms › *refViewSize*—number of nodes contained in the data flow › *refViewError*—description of errors that potentially occurred when calculating the data flow › *initFollowTime*—the time needed to calculate the initial dataflow diagram that is visualized › *initFollowSize*—number of nodes contained in the initial dataflow diagram that is visualized › *isFromPermaLink*—flag whether the origin of the request is a permalink › *scopeMessage*—a message describing how the data flow had to be de-scoped, if it is too large to be visualized › *form*—values of all fields of the form whose sending initiated the dataflow request › *startNodeMap*—starting nodes used for the dataflow analyses › *flowStateGuid*—the technical state of the returned dataflow view
MANTA Flow Viewer	`flow_vis_contractionEvent`	This action is logged when the result of a data lineage analysis is too big to be visualized in MANTA Flow Viewer and has to be contracted.	› *message*—a message describing how the data flow had to be contracted
MANTA Flow Viewer	`flow_vis_aggEdgesIssue`	This action is logged when the result of a data lineage analysis contains too many aggregate edges to be visualized in MANTA Flow Viewer and, as a result, the aggregate edges have to be removed.	N/A
MANTA Flow Viewer	`flow_vis_sourceCode`	This action is logged when the user views the content of any source script in the MANTA Flow Viewer application.	› *node*—the node for which the source code is shown › *resource*—resource of the node for which the source code is shown › *flowStateGuid*—the technical state of the returned dataflow view
MANTA Flow Viewer	`flow_vis_permaLink`	This action is logged when the user generates a permalink for a specific data lineage visualization in the MANTA Flow Viewer application.	› *flowStateGuid*—the technical state of the returned dataflow view
MANTA Flow Viewer	`flow_vis_exportCsv`	This action is logged when the user exports the result of a data lineage analysis to a CSV. This export can be triggered either from the main screen or from the data lineage visualization screen of the MANTA Flow Viewer application.	› *flowStateGuid*—the technical state of the returned dataflow view
MANTA Flow Viewer	`flow_vis_exportPng`	This action is logged when the user exports a specific data lineage diagram to a PNG file. This export can only be triggered from the data lineage visualization screen of the MANTA Flow Viewer application.	› *flowStateGuid*—the technical state of the returned dataflow view
MANTA Flow Viewer	`flow_vis_exportDirect`	This action is logged when the user executes the analysis and export of the data lineage directly from the main screen of the MANTA Flow Viewer application. (The lineage is not visualized in MANTA Flow Viewer in this case.)	The params for this action are the same as for `flow_vis_refView` action.
MANTA Flow Merger	`flow_revision_new`	This action is logged whenever a new (major or minor) revision is created in the MANTA Flow Server repository.	› *revision*—the revision number of the newly created revision
MANTA Flow Merger	`flow_revision_commit`	This action is logged whenever an opened (major or minor) revision is committed in the MANTA Flow Server repository.	› *revision*—the revision number of the committed revision
MANTA Flow Merger	`flow_revision_rollback`	This action is logged whenever an opened (major or minor) revision is rolled back in the MANTA Flow Server repository.	› *revision*—the revision number of the rolled back revision › *result*—result of the rollback operation – *vertexActionMap*—map with operations performed over nodes – *edgeActionMap*—map with operations performed over edges
MANTA Flow Merger	`flow_revision_pruneTo`	This action is logged whenever the content of the MANTA Flow Server repository is pruned from the oldest revision to a specific (newer) revision.	› *perservedRevisionCount*—number of revisions that were preserved by the prune operation › *oldestCommittedRevision*—the oldest revision that was preserved by the prune operation

			› *latestCommittedRevision*—the latest revision that was preserved by the prune operation › *result*—result of the prune operation – *vertexActionMap*—map with operations performed over nodes – *edgeActionMap*—map with operations performed over edges
MANTA Flow Merger	`flow_revision_pruneOldest`	This action is logged whenever the content of the MANTA Flow Server repository is pruned and only a specified number of latest revisions is preserved.	The params for this action are the same as for the `flow_revision_pruneOldest` action.
MANTA Flow Merger	`flow_repository_truncate`	This action is logged whenever all the content of the MANTA Flow Server repository is truncated.	N/A
MANTA Flow Merger	`flow_repository_content`	This action is logged when the user views MANTA Flow Server repository statistics.	› *layers*—number of layer types stored in the MANTA Flow Server repository › *resources*—number of resource types stored in the MANTA Flow Server repository › *nodes*—number of node types stored in the MANTA Flow Server repository › *edges*—number of edge types stored in the MANTA Flow Server repository › *attributes*—number of attributes stored in the MANTA Flow Server repository › *revision*—revision number for which the statistics are counted
MANTA Flow Merger	`flow_repository_backLinks`	This action is logged whenever the backlink phase of the repository postprocessing scenario is executed for a specified MANTA resource.	› *revisionInterval*—the interval of revisions in which the operation is executed › *backLinks*—number of created backlinks
MANTA Flow Merger	`flow_repository_unification`	This action is logged whenever the unification phase of the repository postprocessing scenario is executed.	› *resourceName*—the name of the resource for which the was operation executed › *sources*—predicates used for identification of sources that should be unified › *targets*—predicates used for identification of targets that should be unified › *caseInsensitive*—flag whether to unify nodes without regard to their case › *forceMergeSources*—flag whether the unification of sources should be done even if there are more targets or no targets at all › *forceMergeTargets*—flag whether targets should be unified even if there are more of them › *result* – *unifiedObjects*—number of unified objects – *vertexMap*—map of counts of unified nodes per node type – *edgeMap*—map of counts of unified edges per edge type › *time*—the time needed for the unification
MANTA Flow Merger	`flow_repository_edgePropagate`	This action is logged whenever the edge propagation phase of the repository postprocessing scenario is executed.	› *revision*—revision in which the edges are propagated
MANTA Flow Merger	`flow_repository_contraction`	This action is logged whenever the contraction phase of the repository postprocessing scenario is executed.	› *result* – *vertexMap*—map of counts of contracted nodes per node type – *edgeMap*—map of counts of contracted edges per edge type
MANTA Flow Merger	`flow_repository_interpolation`	This action is logged whenever the lineage interpolation phase of the repository postprocessing scenario is executed.	› *result* – *interpolatedEdgeMap*—map of counts of interpolated edges per edge type – *skippedEdgeMap*—map of counts of skipped edges per edge type – *addedTransformationVertexMap*—map of counts of new transformation nodes per node type – *skippedTransformationVertexMap*—map of counts of skipped transformation nodes per node type – *errorTransformationVertexMap*—map of counts of failed attempts for the creation of new transformation nodes per node type
MANTA Flow	`flow_repository_export`	This action is logged whenever an export of the whole MANTA Flow Server repository is executed.	› *revision*—the revision in which the export is performed

Exporter			› *vertices*—map of counts of exported nodes per node type › *edges*—map of counts of exported edges per edge type

Module Specific Configuration

Configurations for the modules are located in the directory `<MANTA_SERVER_HOME>/webapps/manta-dataflow-server/WEB-INF/classes` under the name `<module-name>-context.xml`.

Merger

- Maximum leaf edges during edge propagations
  - Bean ID `edgePropagationHelper`
  - Property name `maximumLeafEdges`

Usage

MANTA Software does not guarantee 100% coverage of the provided inputs.

Preparing Input Scripts

Many technologies allow the automated extraction of metadata information. However, when this is not possible, MANTA Flow supports manually supplied metadata files placed in a subdirectory of `<MANTA_DIR_HOME>`.

The MANTA_DIR_HOME directory structure is detailed in MANTA Flow Scanner (Client) Configuration.

Default Metadata Input

`ex. mantaflow/cli/input`

Input metadata files may be copied into the static directories for each technology and configured connection ID. Follow the technology-specific definitions below in the Technology Directory Contents section.

The input directory contains directories with the defined technology-specific connection ID and sub-directory structure.

› `ex. input/msssql/${mssql.dictionary.id}[/database_name[/schema_name]]`
› `ex. input/oracle/${oracle.dictionary.id}/[/schema_name]`

Review the technology-specific definitions below.

API Metadata Input

`POST manta-admin-gui/public/process-manager/v1/executions`

Metadata may also be supplied at the time of executing a workflow in the Admin UI, or by calling the `manta-admin-gui/public/process-manager/v1/executions`method in the Administration API. This provides a method of automating the upload of a ZIP file and executing a workflow to process the contained metadata. When processing metadata dynamically using a ZIP file upload, the contents will contain the same directory structure and files as would be supplied to a static input directory.

When providing metadata files using the Admin UI or API, the contents of the ZIP are processed and analyzed in the lineage database. This allows MANTA Flow to ingest multiple uploads with metadata without overwriting files in the default input directories for each technology. Metadata files uploaded to the server are processed for the specific execution only, and automatically deleted when the workflow completes.

The ZIP file must contain a folder named “input” with the defined technology-specific sub-directories containing metadata files.

› `ex. input/msssql/${mssql.dictionary.id}[/database_name[/schema_name]]`
› `ex. input/oracle/${oracle.dictionary.id}/[/schema_name]`

Review the technology-specific definitions below.

Technology Directory Contents

Oracle PL/SQL

By default, MANTA Flow extracts DDL scripts from configured Oracle databases and then analyzes the data flows in them. It is also possible to analyze the data flows in the PL/SQL scripts that work with the configured database.

› Copy the PL/SQL scripts to the `<MANTA_DIR_HOME>/input/oracle/${oracle.dictionary.id}[/schema_name]` folder, where `oracle.dictionary.id` is the identification of the database (if not changed in the configuration) and `schema_name` is an optional directory structure that gives MANTA information about the default schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

Teradata

By default, MANTA Flow extracts DDL scripts from configured Teradata databases and then analyzes the data flows in them. It is also possible to analyze the data flows in the BTEQ and TPT scripts that work with the configured database.

› Copy the BTEQ scripts to the `<MANTA_DIR_HOME>/input/teradata/${teradata.dictionary.id}/bteq[/database_name]` folder, where `teradata.dictionary.id` is the identification of the database (if not changed in the configuration) and `database_name` is an optional directory structure that gives MANTA information about the default database for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.
› Copy the TPT scripts to the `<MANTA_DIR_HOME>/input/teradata/${teradata.dictionary.id}/tpt` folder, where `teradata.dictionary.id` is the identification of the database (if not changed in the configuration).

SQL Server and T/SQL

By default, MANTA Flow extracts DDL scripts from configured MS SQL servers and then analyzes the data flows in them. It is also possible to analyze the data flows in the T-SQL scripts that work with the configured server.

› Copy the T-SQL scripts to the `<MANTA_DIR_HOME>/input/mssql/${mssql.dictionary.id}[/database_name[/schema_name]]` folder, where `mssql.dictionary.id` is the identification of the server (if not changed in the configuration) and `database_name` and `schema_name` are optional directory structures that give MANTA information about the default database and schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

HiveQL

By default, MANTA Flow extracts DDL scripts from configured Hive databases and then analyzes the data flows in them. It is also possible to analyze the data flows in the HiveQL scripts that work with the configured database.

› Copy the HiveQL scripts to the `<MANTA_DIR_HOME>/input/hive/${hive.dictionary.id}[/database_name]` folder, where `hive.dictionary.id` is the identification of the database (if not changed in the configuration) and `database_name` is an optional directory structure that gives MANTA information about the default database for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

Netezza and NZPLSQL

By default, MANTA Flow extracts DDL scripts from configured Netezza servers and then analyzes the data flows in them. It is also possible to analyze the data flows in the Netezza NZPLSQL scripts that work with the configured server.

› Copy the Netezza NZPLSQL scripts to the `<MANTA_DIR_HOME>/input/netezza/${netezza.dictionary.id}[/database_name[/schema_name]]` folder, where `netezza.dictionary.id` is the identification of the server (if not changed in the configuration) and `database_name` and `schema_name` are optional directory structures that give MANTA information about the default database and schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

PostgreSQL, Greenplum, Redshift, Yellowbrick, Amazon RDS, or Amazon Aurora for PostgreSQL and PL/pgSQL

By default, MANTA Flow extracts DDL scripts from configured PostgreSQL, Greenplum, Yellowbrick, Redshift, Amazon RDS, or Amazon Aurora for PostgreSQL servers and then analyzes the data flows in them. It is also possible to analyze the data flows in the PostgreSQL PL/pgSQL scripts, Greenplum scripts, Yellowbrick scripts, Redshift scripts, Amazon RDS, or Amazon Aurora for PostgreSQL scripts that work with the configured server.

› Copy the PostgreSQL, Greenplum, Yellowbrick, Redshift, Amazon RDS, or Amazon Aurora for PostgreSQL PL/pgSQL scripts to the `<MANTA_DIR_HOME>/input/postgresql/${postgresql.dictionary.id}[/database_name[/schema_name]]` folder, where `postgresql.dictionary.id` is the identification of the server (if not changed in the configuration) and `database_name` and `schema_name` are optional directory structures that give MANTA information about the default database and schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

DB2 PL/SQL

By default, MANTA Flow extracts DDL scripts from configured DB2 databases and then analyzes the data flows in them. It is also possible to analyze the data flows in the DB2 PL/SQL scripts that work with the configured database.

› Copy the DB2 PL/SQL scripts to the `<MANTA_DIR_HOME>/input/db2/${db2.dictionary.id}[/schema_name]` folder, where `db2.dictionary.id` is the identification of the database (if not changed in the configuration) and `schema_name` is an optional directory

structure that gives MANTA information about the default schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

Informatica PowerCenter

By default, MANTA Flow automatically extracts Informatica PowerCenter workflows and other required information from the configured repository and then analyzes the data flows in them.

For successful analysis, it is necessary to have all parameter and indirect files referenced by extracted workflows and sessions. Since the parameter files are not stored by Informatica in the PowerCenter repository, it is necessary to provide MANTA with access to these files. If possible, copy these files into a folder structure similar to the system where the client-side application is installed or copy them anywhere, such as `<MANTA_DIR_HOME>/input/ifpc/${ifpc.extractor.repository}/paramFiles`, and set the path to that directory to the property `ifpc.parameter.files.dirs` in the PowerCenter connection definition.

It is also possible to extract workflows, PowerCenter Integration Service settings, and connections manually; for example, for quick test purposes. In such cases:

- Copy all workflow XML files extracted from the Informatica PowerCenter repository to the `<MANTA_DIR_HOME>/temp/ifpc/${ifpc.extractor.repository}/workflow` folder, where `ifpc.extractor.repository` is the name of the repository (if not changed in the configuration). Workflows can be extracted manually using the PowerCenter Repository Manager tool by checking all export options, or in an automated way using Informatica's command line utility pmrep as follows.

```
c:\Informatica\> pmrep
connect -r <repository_name> -h <portal_host_name> -o
<portal_port_number> -n <user_name> -x <password> -s INTERNAL
# e.g. connect -r INFA_REP -h 192.168.0.16 -o 6005 -n Administrator -x
XXX -s SEC_DOMAIN

# To get a list of all available folders, use
listobjects -o Folder
# To get a list of all wokflows in a specific folder, use
listobjects -o Workflow -f <Folder>
# And this command to export a workflow to an XML file:
ObjectExport -o Workflow -n <workflow_name> -f <folder_name> -u
<xml_file_path> -m -s -b -r -l pmrep.log
```

- Extract the PowerCenter Integration Service settings as described in Informatica PowerCenter Resource Configuration in the section Integration Service Settings and place them in `<MANTA_DIR_HOME>/temp/ifpc/${ifpc.extractor.repository}/ifpcServiceSettings.prm`.
- Extract the PowerCenter Connection details as described in Informatica PowerCenter Resource Configuration in the section Connection Definition Settings and place them in `<MANTA_DIR_HOME>/temp/ifpc/${ifpc.extractor.repository}/ifpcConnectionDefinition.prm`. PowerCenter's command line utility pmrep can be used to automate this process if needed.

```
c:\Informatica\> pmrep
connect -r <repository_name> -h <portal_host_name> -o
<portal_port_number> -n <user_name> -x <password> -s INTERNAL
# e.g. connect -r INFA_REP -h 192.168.0.16 -o 6005 -n Administrator -x
XXX -s SEC_DOMAIN

# To get a list of all available connections, use
ListConnections
# To get details of a specific connection, use
GetConnectionDetails -n <Name> -t <Type>
```

- Disable the `ifpcWorkflowExtractor` step in `<MANTA_DIR_HOME>/cli/scenarios/manta-dataflow-cli/bin/_run_extract.[bat|sh]` or Process Manager.

SSIS

By default, MANTA Flow extracts SSIS packages and projects from the configured repository and then analyzes the data flows in them. It is also possible to provide SSIS packages and projects directly to MANTA. In this case:

- Copy all DTSX, DTSCONFIG, and ISPAC files extracted from the repository to the `<MANTA_DIR_HOME>/temp/ssis/${ssis.extractor.server}` folder, where `ssis.extractor.server` is the name of the repository (if not changed in the configuration).
- Disable the `ssisExtractor` step in `<MANTA_DIR_HOME>/cli/scenarios/manta-dataflow-cli/bin/_run_extract.[bat|sh]` or Process Manager.

For successful analysis, it is necessary to have all parameter and indirect files referenced by extracted workflows and sessions. Copy these files into a folder structure similar to the system where the client-side application is installed.

SSAS

To analyze SSAS project files:

- Copy all XMLA or BIM files extracted from SSAS to the `<MANTA_DIR_HOME>/input/ssas/${ssas.dictionary.id}` folder, where `ssas.dictionary.id` is the identification of the server (if not changed in the configuration).

Talend

To analyze Talend project files:

- Copy all Talend XML jobs to the `<MANTA_DIR_HOME>/input/talend/${talend.system.id}` folder, where `talend.system.id` is the identification of the project (if not changed in the configuration).

DataStage

To analyze DataStage project files:

- Copy all parallel DataStage jobs you want to analyze and the parameter sets you use in those jobs as one XML file, exported from Designer Client in the *Repository Export* window to the `<MANTA_DIR_HOME>/temp/datastage/${datastage.extractor.server}` folder, where `datastage.extractor.server` identifies both the DataStage connection in Manta and the project name in DataStage (if not changed in the configuration).

For successful analysis, it may be necessary to also provide MANTA with additional parameter and indirect files. Put these files in the folders defined in the configuration properties according to the Source System Requirements section in DataStage Resource Configuration.

Pig Latin

To analyze Pig Latin scripts:

- Copy all Pig Latin scripts to the `<MANTA_DIR_HOME>/input/piglatin/${piglatin.dictionary.id}/script` folder, where `piglatin.dictionary.id` is the identification of the Pig Latin system (if not changed in the configuration).
- Copy all Pig Latin macros used in Pig Latin scripts to the `<MANTA_DIR_HOME>/input/piglatin/${piglatin.dictionary.id}/macro` folder, where `piglatin.dictionary.id` is the identification of the Pig Latin system (if not changed in the configuration).

Sqoop

To analyze Sqoop scripts not stored in the Sqoop repository:

- Copy all Sqoop scripts to the `<MANTA_DIR_HOME>/input/sqoop/${sqoop.system.id}` folder, where `sqoop.system.id` is the identification of the Sqoop repository (if not changed in the configuration).

Excel

To analyze Excel workbooks:

- Copy all Excel workbooks in XLSX or XLSM to the `<MANTA_DIR_HOME>/input/excel/${excel.dictionary.id}` folder, where `excel.dictionary.id` is the identification of the Excel system (if not changed in the configuration).

ER/Studio

To analyze ER/Studio models:

- Copy all ER/Studio models to the `<MANTA_DIR_HOME>/input/erstudio/${erstudio.system.id}/models` folder, where `erstudio.system.id` is the identification of the ER/Studio system (if not changed in the configuration).
- Put the `connections.ini` file in the described format in the `<MANTA_DIR_HOME>/input/erstudio/${erstudio.system.id}` folder, where `erstudio.system.id` is the identification of the ER/Studio system (if not changed in the configuration).

PowerDesigner

To analyze PowerDesigner models:

- Copy all PowerDesigner models in CDM, LDM, or PDM to the `<MANTA_DIR_HOME>/input/powerdesigner/${powerdesigner.system.id}/models` folder, where `powerdesigner.system.id` is the identification of the PowerDesigner system (if not changed in the configuration).
- Put the `connections.ini` file in the described format in the `<MANTA_DIR_HOME>/input/powerdesigner/${powerdesigner.system.id}` folder, where `powerdesigner.system.id` is the identification of the PowerDesigner system (if not changed in the configuration).

Cobol/JCL

To analyze Cobol/JCL scripts:

- Copy all Cobol and JCL scripts to the `<MANTA_DIR_HOME>/input/cobol/${cobol.dictionary.id}` folder, where `cobol.dictionary.id` is the identification of the Cobol/JCL scripts system (if not changed in the Cobol common configuration properties).
- Copy all Cobol copybooks used with Cobol scripts to the `<MANTA_DIR_HOME>/input/cobol/${cobol.dictionary.id}/${cobol.script.copybookDir}` folder, where `cobol.dictionary.id` is the identification of the Cobol/JCL scripts system (if not changed in the Cobol common configuration properties) and `cobol.script.copybookDir` is the related directory of Cobol copybooks.
- If the Cobol scripts use embedded SQL, create the prm connection definition file `connectionsConfiguration.prm` in `<MANTA_DIR_HOME>/input/cobol/${cobol.dictionary.id}` (if not changed in the Cobol common configuration properties). The prm file should contain one connection definition named DEFAULT with valid connection properties. For more information about this file and its format, see the section on connection definition settings in Informatica PowerCenter Resource Configuration.

ODI

By default, MANTA Flow extracts ODI metadata from the configured ODI repository and then analyzes the data flows in it. It is also possible to analyze the data flows in the manually-provided Smart Exports of ODI objects exported from the ODI Studio.

To analyze manually-provided ODI Smart Exports:

- Copy all ODI Smart Exports exported from ODI Studio to the `<MANTA_DIR_HOME>/temp/odi/${odi.extractor.repository}` folder, where `odi.extractor.repository` is the identification of the ODI repository (if not changed in the ODI common configuration properties).
- Copy all ODI Exports from the execution environment to the `<MANTA_DIR_HOME>/temp/odi/${odi.extractor.repository}/technologies` folder, where `odi.extractor.repository` is the identification of the ODI repository (if not changed in the ODI common configuration properties). These exports have to start with the "TECH_" prefix.
- For more information on how to extract exports from ODI Studio, see the guide on How to Extract Exports from ODI Studio.

Snowflake

By default, MANTA Flow extracts DDL scripts from the configured Snowflake server and then analyzes the data flows in them. It is also possible to analyze the data flows in the Snowflake SQL scripts that work with the configured server.

- Copy the Snowflake SQL scripts to the `<MANTA_DIR_HOME>/input/snowflake/${snowflake.dictionary.id}[/database_name[/schema_name]]` folder, where `snowflake.dictionary.id` is the identification of the server (if not changed in the configuration) and `database_name` and `schema_name` are optional directory structures that give MANTA information about the default database and schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

StreamSets

By default, MANTA Flow extracts StreamSets pipelines from the configured StreamSets Data Collector (SDC) server and then analyzes the data flows in them. It is also possible to analyze the data flows in the manually-provided StreamSets pipelines exported from the SDC.

- Copy all StreamSets pipelines exported from the SDC in JSON file format to the `<MANTA_DIR_HOME>/temp/streamsets/${streamsets.extractor.server}` folder, where `streamsets.extractor.server` is the identification of the SDC (if not changed in the StreamSets common configuration properties).

BigQuery

By default, MANTA Flow extracts DDL scripts from the configured BigQuery server and then analyzes the data flows in them. It is also possible to analyze the data flows in the BigQuery SQL or job scripts that work with the configured server.

- Copy the BigQuery SQL scripts to the `<MANTA_DIR_HOME>/input/bigquery/${bigquery.dictionary.id}/sql[/project_id[/dataset_name]]` folder, where `bigquery.dictionary.id` is the identification of the configured connection (if not changed in the configuration) and `project_id` and `dataset_name` are optional directory structures that give MANTA information about the default project and schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.
- Copy the BigQuery job scripts to the `<MANTA_DIR_HOME>/input/bigquery/${bigquery.dictionary.id}/jobs` folder, where `bigquery.dictionary.id` is the identification of the configured connection (if not changed in the configuration).

SAP HANA

By default, MANTA Flow extracts DDL scripts from configured SAP HANA cloud or SAP HANA on premise database and then analyzes the data flows in them. It is also possible to analyze the data flows in the SAP HANA SQL scripts that work with the configured database.

› Copy the SAP HANA SQL scripts to the `<MANTA_DIR_HOME>/input/saphana/${saphana.dictionary.id}[/schema_name]` folder, where `saphana.dictionary.id` is the identification of the configured database connection (if not changed in the configuration) and `schema_name` is an optional directory structure that gives MANTA information about the default schema for any unqualified references to database objects used in the scripts. See the section *Unqualified Object References* below.

Python

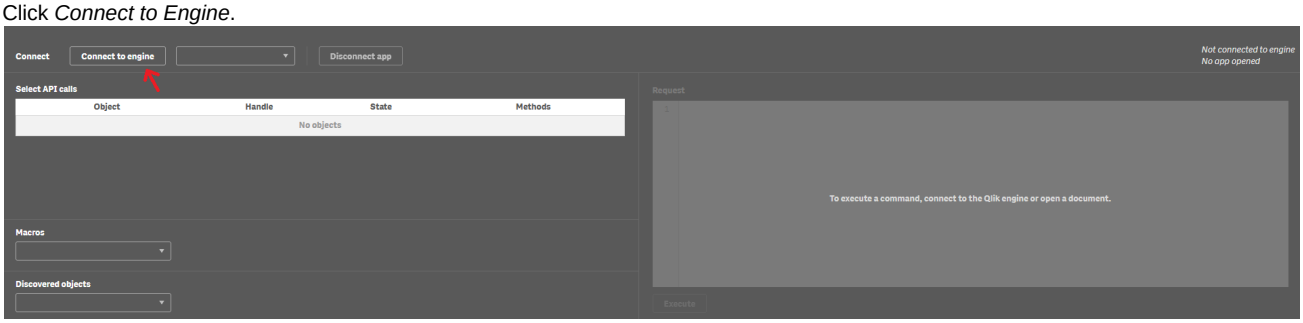
By default, MANTA Flow extracts Python programs from a directory or a ZIP archive. An absolute path to the directory or the archive which should be extracted has to be provided when configuring a new Python connection. Make sure that the extracted directories or archives are in a location accessible by MANTA (therefore no Administrator or other special access is required).

› Place the whole analyzed Python program into a single directory or create a ZIP archive containing all files.
› When setting up the connection as described in Python Resource Configuration, use the absolute path to the directory or a ZIP archive in the `python.system.application.path` property.

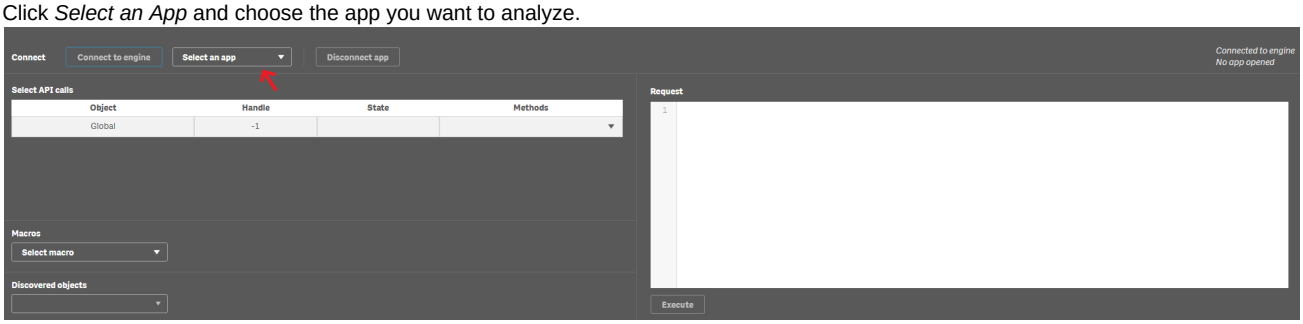
Qlik Sense

By default, MANTA Flow extracts Qlik Sense applications from the Qlik Sense server. It is also possible to extract an application manually as follows.

1. Go to `<server address>/dev-hub/engine-api-explorer`.
2. Click *Connect to Engine*.



3. Click *Select an App* and choose the app you want to analyze.



4. Open the developer console in your browser.
   › In Chrome, Firefox, or Edge, click `CTRL+SHIFT+I` and select the *Console* tab.
5. Download the following script, paste its content into the console, and press *Enter*.

Once the script finishes, the file `output.zip` should be downloaded on your computer. Extract the content of the zip file into`<MANTA_DIR_HOME>/input/qliksense/${qliksense.extractor.server}/<application name>`, where `qliksense.extractor.server` is the identification of the Qlik Sense server. If you want to extract more applications, **refresh the page** and repeat the steps described above.

Kafka

The Confluent Schema Registry schemas can be extracted either automatically or manually (as of R37) as follows.

1. Download the Schema Registry schema using the Schema Registry REST API.
   a. List all the subjects: `kafka.schema.registry.com:8081/subjects/`.
   b. For each subject, list its versions: `kafka.schema.registry.com:8081/subjects/<subject_name>/versions`.
   c. List the version(s) you want to use: `kafka.schema.registry.com:8081/subjects/<subject_name>/versions/<version_number>`.
   d. Save the response to the last request in a file. (Each schema has to be in a separate file.)

```
{
    "subject": "avro-registration-value",
    "version": 1,
    "id": 12,
    "schema": "{\"type\":\"record\",\"name\":\"
KsqlDataSourceSchema\",\"namespace\":\"io.confluent.ksql.
avro_schemas\",\"fields\":[{\"name\":\"registertime\",\"type\":[\"
null\",\"long\"],\"default\":null},{\"name\":\"userid\",\"type\":[\"
null\",\"string\"],\"default\":null},{\"name\":\"regionid\",\"
type\":[\"null\",\"string\"],\"default\":null},{\"name\":\"gender\",
\"type\":[\"null\",\"string\"],\"default\":null}]}"
}
```

2. Place all the schemas in `<MANTA_DIR_HOME>/input/kafka/<kafka_connection_id>/schema_registry/`.

Fivetran

Fivetran resources have to be exported and provided to MANTA for dataflow analysis; see the section on manual Fivetran export below. To analyze Fivetran files, copy the exported Fivetran resources to the `<MANTA_DIR_HOME>/input/fivetran/${fivetran.extractor.server}` folder, where `fivetran.extractor.server` is the *Fivetran Server Name* from Admin UI.

There are two groups of resources that need to be exported.

1. Destination metadata files
2. SQL transformation project

Destination Metadata Files

These files can only be extracted through Fivetran API.

- List all groups endpoint: `https://api.fivetran.com/v2/groups`
  - This endpoint returns an array of all destinations within the Fivetran instance. Every element of the array has an:
    - id—ID of the destination
    - name—name of the destination from Fivetran UI
  - Destination configuration endpoint: `https://api.fivetran.com/v1/destinations/{destination id from "List all groups" endpoint}`
    - Save the results of this endpoint in a file with a `.json` extension and add it to the destination folder. The structure is described in Fivetran Integration Requirements.
  - List all connectors within the destination endpoint: `https://api.fivetran.com/v1/groups/{destination id from "List all groups" endpoint}/connectors`
    - This endpoint returns an array of items (connectors). Every element of the array has an:
      - id—ID of the connector
    - You will need to use this endpoint for every destination you want to extract.
  - Connector configuration endpoint: `https://api.fivetran.com/v1/connectors/{connector id from "List Connectors within the destination" endpoint}`
    - Save the results of this endpoint in a connector configuration file with a `-conf.json` extension and add it to a folder for the particular connector. The structure is described in Fivetran Integration Requirements.
    - You will need to use this endpoint for every connector you want to extract within the destination.
  - Connector schema endpoint: `https://api.fivetran.com/v1/connectors/{connector id from "List Connectors within the destination" endpoint}/schemas`
    - Save the results of this endpoint in a connector schema file with a `.json` extension and add it to a folder for the particular connector. The structure is described in Fivetran Integration Requirements.
    - You will need to use this endpoint for every connector you want to extract within the destination.

API Authorization

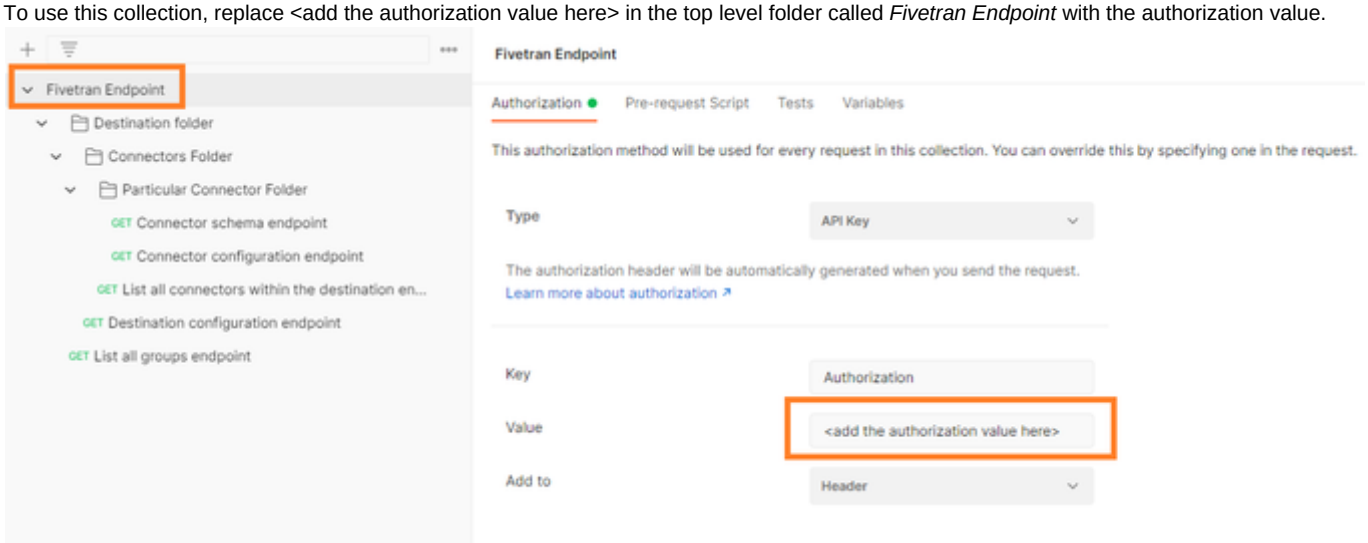
Fivetran uses API Key authentication. The header of every request to the API must contain an authorization key with a value set to `Basic {api_key}:{api_secret}`, where `{api_key}:{api_secret}` have to be encoded in base64. The API key and secret can be generated through Fivetran UI by a user with the *Account Administrator*(new role model) / *Owner*(legacy role model) role.

A detailed description of how to generate a key and secret can be found on the official Fivetran website: https://fivetran.com/docs/rest-api/getting-started.

A detailed description of role permissions can be found on the official Fivetran website: https://fivetran.com/docs/getting-started/fivetran-dashboard/account-management/role-based-access-control.

The Postman Collection with Endpoints for Easier Extraction

To use this collection, replace <add the authorization value here> in the top level folder called *Fivetran Endpoint* with the authorization value.



SQL Transformation Project

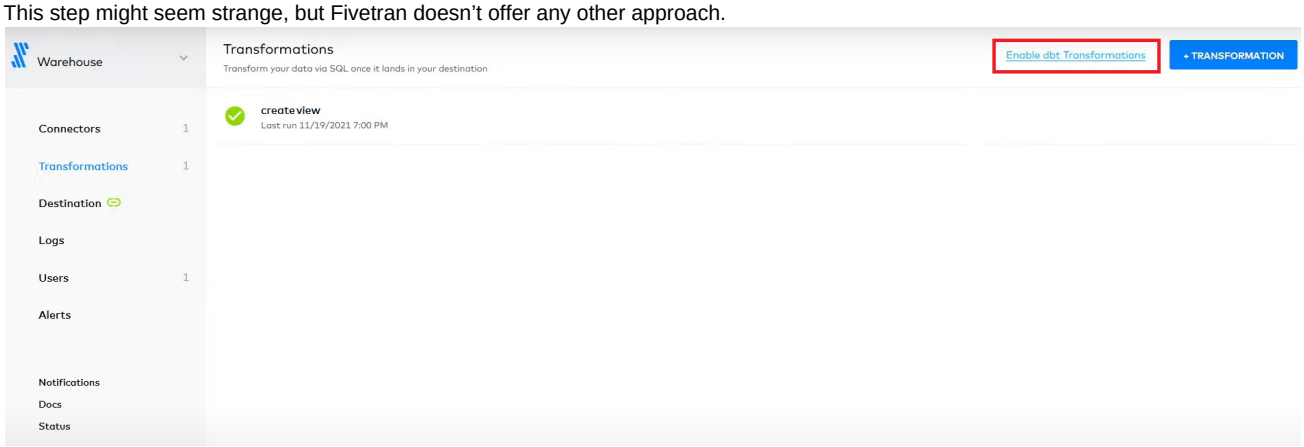
Fivetran supports two types of SQL projects.

1. Basic transformation project
2. Transformations for dbt Core

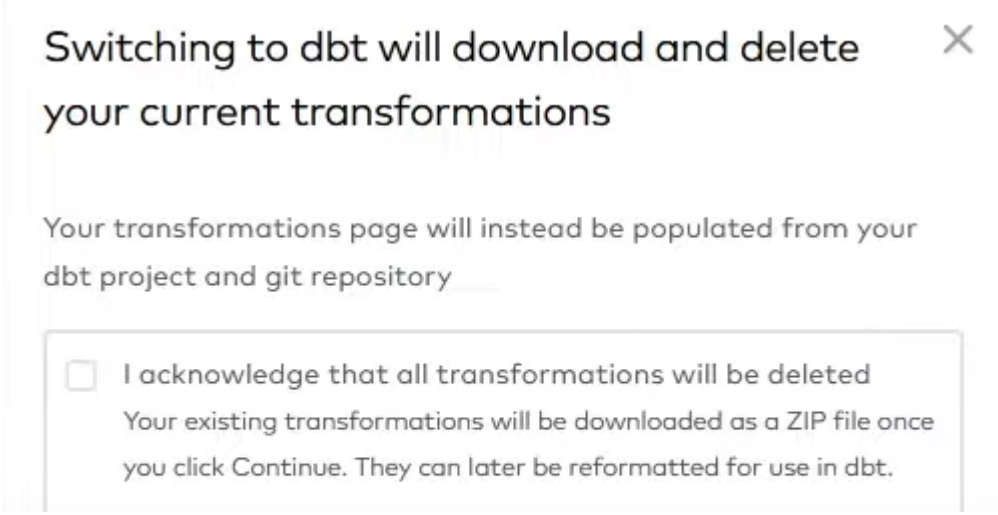
Basic Transformation Project

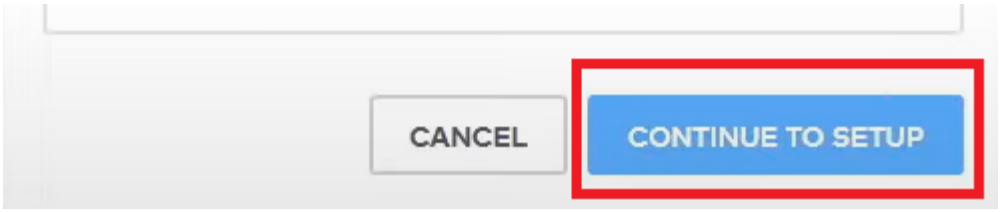
The basic transformation project can be extracted through *Transformation* tab in the destination dashboard in Fivetran UI.

› Click on *Enable dbt Transformation*.
  – This step might seem strange, but Fivetran doesn't offer any other approach.

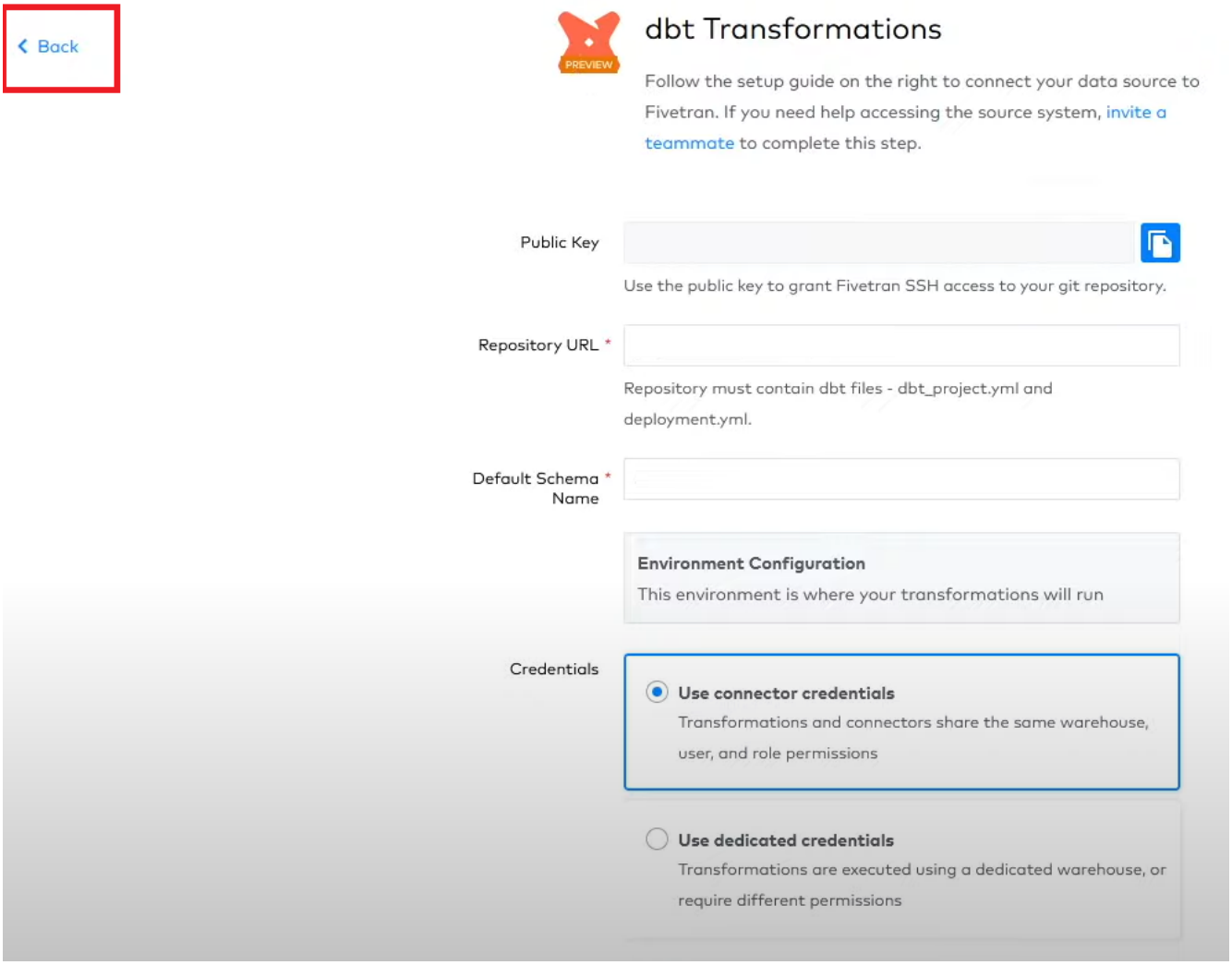


› After clicking on the button, a window will appear.





› Click on *Continue to Setup*, to start downloading the basic transformations.



› Abort the transfer to the DBT transformations by clicking on the back button in the upper left corner.

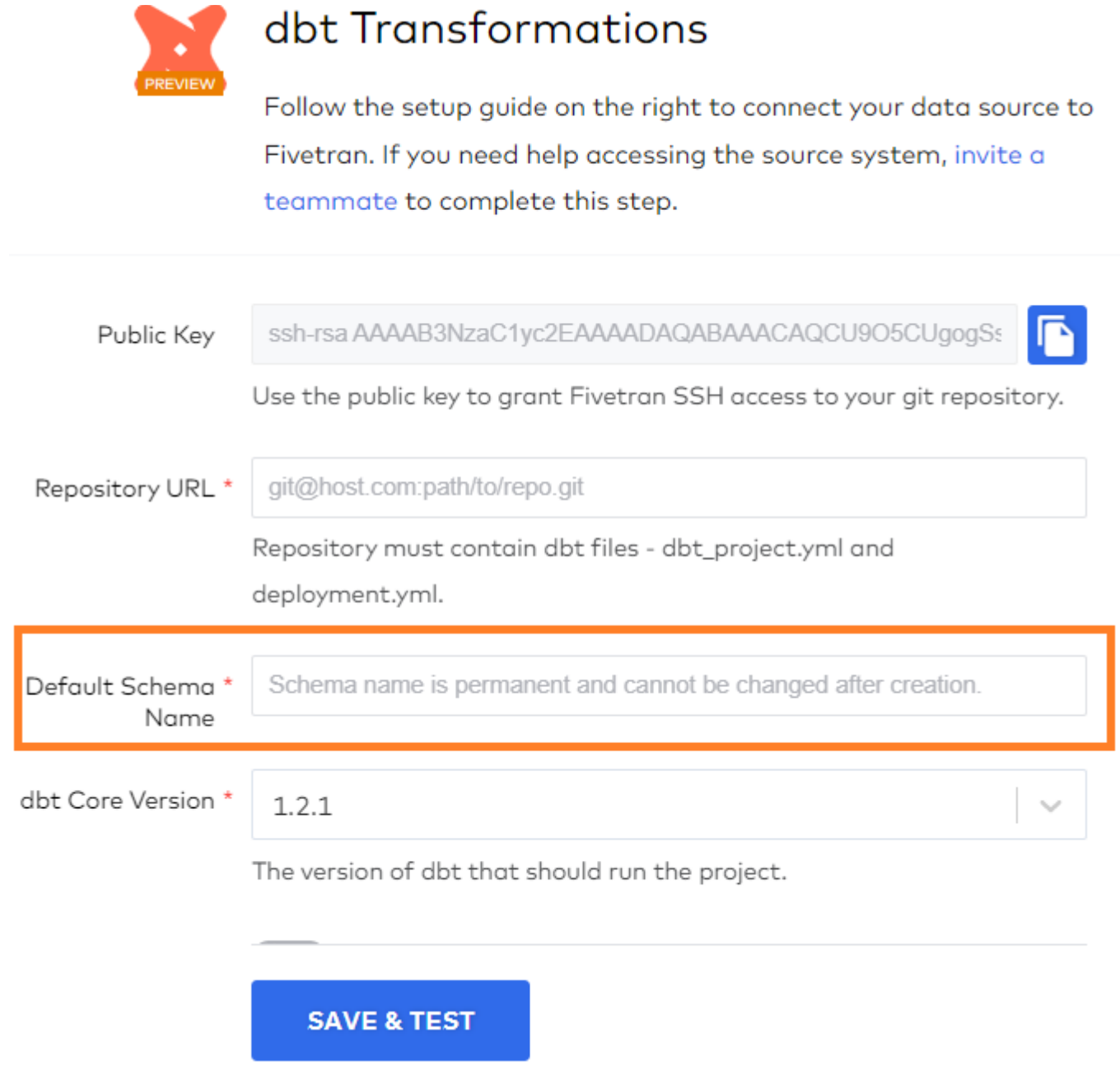
Transformations for dbt Core

Transformations for the dbt project can be extracted with a series of two commands.

› `dbt compile`
  – This command compiles all SQL queries in the dbt project. Without executing this command, the documentation generated by the following command won't contain the compiled query.
› `dbt docs generate`
  – This command generates documentation for previously executed commands: in our case, for the *dbt compile*.
  – The generated documentation contains multiple files, but only the *manifest.json* file is needed. Save this file in the structure described in Fivetran Integration Requirements.

Schema Names

The results of DBT queries are saved in a schema named `<target_schema>_<custom_schema>`. *Custom Schemas* are present in the manifest file and no further steps are required, but that's not the case for *Target Schemas*. For the scanner to work correctly, it is necessary to generate documentation with an account that has the *target_schema* set to the same value as the *Default Schema Name*, which is set in Fivetran UI when connecting the DBT project.



The target schema can be set in `profiles.yml` as described on the official website https://docs.getdbt.com/reference/profiles.yml for the CLI version of dbt.

Azure Data Factory

ADF resources have to be exported and provided to MANTA for the dataflow analysis; see the section on manual ADF export below. To analyze ADF files, copy the exported ADF resources to the `<MANTA_DIR_HOME>/input/datafactory/${datafactory.system.id}` folder, where `datafactory.system.id` is the ID of the connection.

The following three methods can be used to export ADF resources.

1. Export from Git via command line *(recommended, allows for automation of the extraction)*
2. Manual download from GitHub (Exports all resources at once) *(recommended)*
3. Manual download from ADF (Exports only one resource at a time)

Export from Git via Command Line

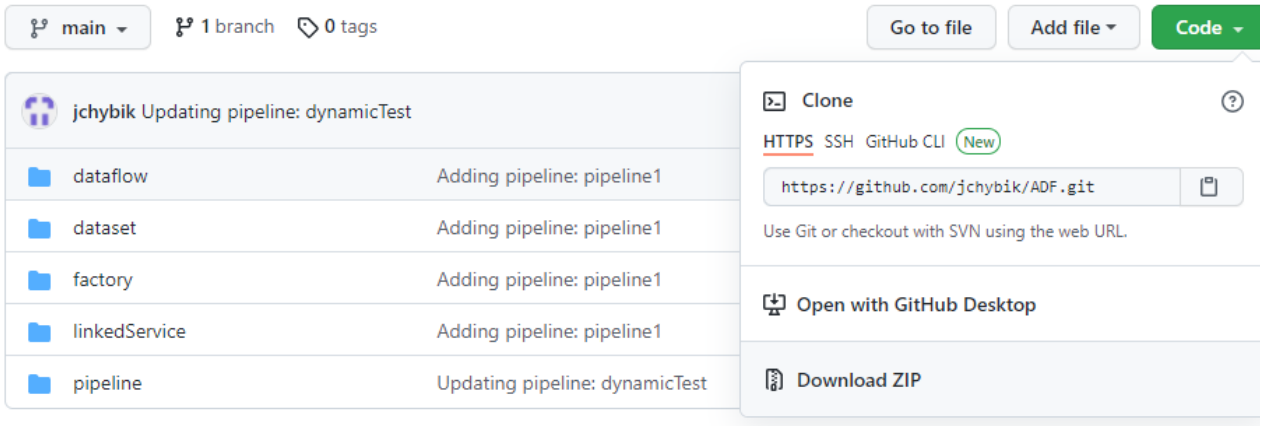
This method exports all ADF resources at once.

- *To export* all ADF resources at once, run the command `git clone <ADF repository URL> <Connection folder>` where `<ADF repository URL>` is the repository containing the ADF resources and `<Connection folder>` is the connection folder that should contain all resources, which by default is `${manta.dir.input}/datafactory/${datafactory.system.id}`.
- *To update* an existing connection exported this way, run the command `git -C <Connection folder> pull`.
- Running the `git -C` command prior to each dataflow analysis is the best way to automate the extraction. Both initial `git clone` and `git -C` commands must be triggered outside MANTA.

Manual Download from GitHub

This method exports all ADF resources at once.

1. In the GitHub repository, click the *Code* button.

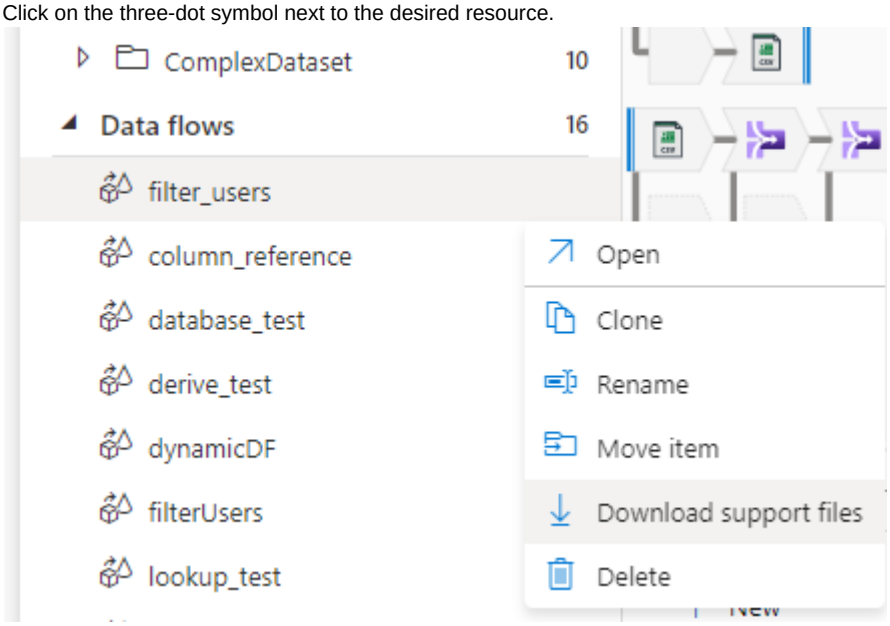


2. Select *Download ZIP*.
3. Extract the contents into the folder `${manta.dir.input}/datafactory/${datafactory.system.id}`. (Do not place the ZIP file directly there.)

Manual Download from ADF

This method exports one resource at a time, together with the resources it is dependent on.

1. Click on the three-dot symbol next to the desired resource.



2. Select *Download Support Files*.
3. Extract the contents into the folder `${manta.dir.input}/datafactory/${datafactory.system.id}`. (Do not place the ZIP file there directly.)

Unqualified Object References

Objects (tables, views, procedures, functions, etc.) used in SQL queries can be fully, semi, or unqualified.

- An example of a fully-qualified reference to the table/view `customer` where the database (`dwh_db`) and schema (`core`) are specified when referencing the table in the query:

```
SELECT * FROM dwh_db.core.customer
```

  Please note that Oracle and Teradata do not support this three-level hierarchy. See the next bullet point.
- An example of a semi-qualified reference to the table/view `customer` where the schema (`core`) is specified when referencing the table in the query but the database specification is missing. The current database will be used when executing the query.

```
SELECT * FROM core.customer
```

  Please note that for Teradata and Oracle this syntax would actually represent a fully-qualified reference as Teradata does not have schemas. (Objects are placed directly in a database.) Oracle uses different syntax (DB Links) to reference another database.
- An example of an unqualified reference to the table/view `customer` where no schema or database is specified. The current database and schema will be used when executing the query.

```
SELECT * FROM customer
```

Since the scripts are provided to MANTA as files, MANTA also needs information about the *current* schema and/or database name against which the semi- and unqualified object references should be evaluated to make sure the lineage connects properly.

Running a Data Flow Analysis

MANTA Flow components must be launched before the dataflow analysis starts. As of R35, MANTA Launcher is available to start all required services.

You can check whether the server side is running by opening this URL in your web browser.

- URL: http://localhost:8080/manta-dataflow-server/api/inspect
- Default administrator name: `admin`
- Default administrator password: `admin`

You should see simple statistics about the metadata repository. Note that the base URL can be changed in the configuration.

The client side of MANTA Flow can be launched by any user with the correct rights to the following application folders.

- Read rights for the `<MANTA_DIR_HOME>/input` folder
- Read/write rights for the `<MANTA_DIR_HOME>/output`, `<MANTA_DIR_HOME>/temp`, and `<MANTA_DIR_HOME>/log` folders

There are multiple ways to launch the lineage analysis.

1. Via Process Manager in Admin UI; see MANTA Process Manager
2. Via Orchestration API; see MANTA Orchestration API
3. As of R36, via bat and shell script wrappers for Orchestration API; see MANTA Orchestration API Scripts
4. Up to R38, via bat and shell scripts for the CLI application (Please note that this is a deprecated method and the scripts will not be available as of MANTA R39. The replacement for this functionality is the shell script wrappers for Orchestration API mentioned in the previous step.)
   - For MANTA standalone (native UI)
     - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run.sh`
     - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run.bat`
   - For MANTA integrated with Alation
     - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run_alation.sh`
     - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run_alation.bat`
   - For MANTA integrated with Collibra
     - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run_colibra.sh`
     - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run_collibra.bat`
   - For MANTA integrated with IBM IGC
     - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run_igc.sh`
     - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run_igc.bat`

- For MANTA integrated with Informatica EDC
  - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run_iedc.sh`
  - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run_iedc.bat`
- For MANTA integrated with Top Quadrant
  - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run_topquadrant.sh`
  - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run_topquadrant.bat`
- For Open MANTA Integration Export
  - Unix: bash `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/_run_openexport.sh`
  - For Windows: `<MANTA_DIR_HOME>\scenarios\manta-dataflow-cli\bin\_run_openexport.bat`

- The batch files can be customized to exclude unused steps, such as unused extractors, by commenting them out to reduce the number of log files generated and the amount of runtime.

Reading Outputs

Some or all of the following features may or may not be accessible based on your MANTA Flow license.

Data Flow Viewer

Now data flows can be viewed using the MANTA Flow Viewer application which is bundled with the server side of the product. Just open the web browser at:

- URL: http://localhost:8080/manta-dataflow-server/viewer
- Default user name: `user`
- Default user password: `manta`

Note that the base URL can be changed in the configuration.

CSV Files

All data flows are also exported to CSV files stored in the `<MANTA_DIR_HOME>/output/csv` folder with the same structure as defined in MANTA Flow Usage: CSV Files and Open MANTA Extensions Usage.

Scheduling a Data Flow Analysis Run

The metadata repository on the server side of MANTA Flow uses a versioning system with transactional processing. (Note that some deployments use a non-transactional version of the repository; this section does not apply to them.) All modifications to the repository during an analysis run are performed within a transaction and are only visible once the transaction is committed.

No more than one transaction may be open at any time. If an analysis is launched while a transaction is still open, the analysis will fail. This may happen because the previous scheduled analysis has not finished running yet (such a situation will be resolved once the previous analysis run finishes and does not require the administrator's intervention) or because the previous run was interrupted before completion (e.g., by an unexpected system shutdown or by an operator's action)—in which case, administrator intervention is necessary and all analysis runs will fail until the administrator resolves the issue by manually performing the transaction rollback.

Considerations When Scheduling an Analysis Run

- Data Flow Viewer users will be able to use the Data Flow Viewer even if an analysis is underway. This makes it possible to schedule an analysis run during working hours. The users may, however, encounter longer server response times due to metadata repository locking by the analysis run. Also, try to schedule the run in such a way that it finishes outside working hours.
- You should allow a run enough time to complete before another scheduled run is started.
- Configure your scheduler to notify the administrator if the analysis run fails (returns a non-zero return value - $? in Unix, %ERRORLEVEL% in Windows). It is essential for the administrator to check the reason for every run failure and determine whether any administrative action is needed.

Export User Documentation

This document introduces the format of metadata exported from the MANTA Flow application and how it can be used for further metadata analysis.

Export Execution

The export can be executed using scripts `<MANTA_HOME>/cli/scenarios/manta-dataflow-cli/bin/exportRepositoryScenario.[bat|sh]`.

As of MANTA 3.30.1, it is also possible to specify which revision of the MANTA repository should be exported (example command: '`exportRepositoryScenario.bat 3.000009'`). The format of the revision number is X.YYYYYY where X is the major revision number and YYYYYY is the minor revision number offset by zeros. If the revision number is not set, the export is executed for the latest revision in the MANTA repository.

Metadata Format

Exported metadata can be stored in CSV files or as tables in a database. This is its database schema.

- DF_Layer—contains the names of metadata layers (i.e., physical, logical, business)
  - ID—unique ID of the layer used as a foreign key for the resources
  - Layer name—name of the layer
  - Layer type—type of layer
- DF_Resource—names and descriptions of resources (i.e., Teradata, Teradata DDL)
  - Resource_Id—the unique ID of the resource used as a foreign key for nodes and edges
  - Resource_Name—the name of the resource
  - Resource_Type—the type of the resource
  - Resource_Description—a description of the resource
  - Layer ID—ID of the layer which this resource belongs to
- DF_Node—nodes from a dataflow graph (e.g., tables, columns, packages, statements, etc.)
  - Node_Id—the unique ID of the node used as a foreign key for edges and node attributes
  - Parent_Node_Id—the ID of the parent node (e.g., columns have tables, packages have schemas) or an empty string if the node is at the top of the hierarchy (e.g., databases, folders)
  - Node_Name—the name of the node
  - Node_Type—the type of the node (all types can be found here)
  - Resource_Id—the ID of the resource the node belongs to
- DF_Edge—edges from a dataflow graph on the lowest possible level (attributes, columns, parameters, etc.)
  - Edge_Id—the unique ID of the edge used as a foreign key for edge attributes
  - Source_Node_Id—the ID of the source node
  - Target_Node_Id—the ID of the target node
  - Edge_Type—the type of the edge
    - DIRECT—a direct data flow (e.g., `insert into target (direct) select direct from source;`)
    - FILTER—a filter data flow (e.g., `insert into target select * from source where filter = 0;`)
    - MAPS_TO—maps nodes from different layers (e.g., maps *First Name* attribute [business/logical layer] to *NAME_FIRST* column [technical layer])
  - Resource_Id—the ID of the resource the edge belongs to
- DF_Node_Attribute—further attributes of nodes
  - Node_Id—the ID of the node that the attribute belongs to
  - Attribute_Name—the attribute name
  - Attribute_Value—the attribute value
- DF_Edge_Attribute—further attributes of edges
  - Edge_Id—the ID of the edge the attribute belongs to
  - Attribute_Name—the attribute name
  - Attribute_Value—the attribute value
- DF_Source_Code—source code associated with script nodes; this file is generated only when `manta.exporter.includeSource` is set to `true`
  - Node_Id—the ID of the node the source code belongs to
  - Script_Code—the source code text

Metadata Analysis

This section contains some queries that can be useful for various metadata analyses. Note that all queries are written in Teradata SQL dialect. Also note that all of these queries are provided as **examples** of how the exports can be queried.

Finding a Fully-Qualified Object

When it is necessary to find an object defined by a fully-qualified name, self-join the *DF_Node* table with the *Parent_Node_Id* property. The object resource name is also necessary. For example, it is possible to find the column *ExampleDB.ExampleTable.ExampleColumn* in a Teradata resource using this query.

```
select Col.*
from DF_Node Col
  inner join DF_Node Tab on Col.Parent_Node_Id = Tab.Node_Id
```

```
  inner join DF_Node Dat on Tab.Parent_Node_Id = Dat.Node_Id
  inner join DF_Resource Res on Dat.Resource_Id = Res.Resource_Id
where
  Res.Resource_Name = 'Teradata' and
  Dat.Node_Name = 'ExampleDB' and
  Tab.Node_Name = 'ExampleTable' and
  Col.Node_Name = 'ExampleColumn'
;
```

Here is the result.

```
3      2      ExampleColumn Column 1
```

Listing All Object Attributes

When the object ID is known (e.g., from a query for finding a fully-qualified object), it is easy to list all its attributes using this query.

```
select Atr.Attribute_Name, Atr.Attribute_Value
from DF_Node_Attribute Atr
where Atr.Node_Id = 3
;
```

Here is the result.

```
ORDER                1
COLUMN_CHARSET       UNICODE
COLUMN_LENGTH        50
COLUMN_TYPE          VARCHAR
```

Finding All Direct Flows from an Object (One Step)

When the object ID is known (e.g., from a query for finding a fully-qualified object), it is possible to find all objects that a direct flow leads to (from the identified object) in one step. Note that MANTA exports edges on the attribute level, meaning that flow exists between attribute-level nodes—not between any other objects (such as tables).

```
select Tgt.*
from DF_Edge Edg
inner join DF_Node Src on Src.Node_Id = Edg.Source_Node_Id
inner join DF_Node Tgt on Tgt.Node_Id = Edg.Target_Node_Id
where
Edg.EdgeType = 'DIRECT' and
Src.Node_Id = 3
;
```

Here is the result.

```
6      5      1 ExampleColumn      BTEQ ColumnFlow     2
9      8      1 ExampleColumn      BTEQ ColumnFlow     2
```

Finding All Direct Flows from an Object (Recursive)

If it is necessary to find all direct flows from an object recursively, a recursive statement like this can be created.

```
with recursive Temp_Node(Node_Id, Depth) as (
  select Col.Node_Id, cast(0 as integer) as Depth
  from DF_Node Col
    inner join DF_Node Tab on Col.Parent_Node_Id = Tab.Node_Id
    inner join DF_Node Dat on Tab.Parent_Node_Id = Dat.Node_Id
    inner join DF_Resource Res on Dat.Resource_Id = Res.Resource_Id
  where
    Res.Resource_Name = 'Teradata' and
    Dat.Node_Name = 'SourceDB' and
    Tab.Node_Name = 'SourceTable' and
    Col.Node_Name = 'SourceColumn'
union all
  select Tgt.Node_Id, Src.Depth + 1
  from Temp_Node Src
    inner join DF_Edge Edg on Edg.Source_Node_Id = Src.Node_Id
    inner join DF_Node Tgt on Edg.Target_Node_Id = Tgt.Node_Id
  where
    Edg.Edge_Type = 'DIRECT' and
    Src.Depth <= 10
)
select distinct Node_Id
from Temp_Node;
;
```

However, this statement must be limited to a maximum number of steps (depth). Note that even this limitation will not necessarily prevent the statement from taking a very long time! Therefore, it is better to write the same function using a stored procedure (which also must be limited) or a recursive BTEQ script.

Repository API

Introduction

MANTA Flow Repository API is an HTTP-based interface for MANTA Flow Server.

An interactive OpenAPI (Swagger) is available at the URL `http(s)://<MANTA_SERVER_BASE_URL>/swagger-ui/`. Non-interactive documentation is available at MANTA Flow Repository API REST.

Authentication and Authorization

The repository API can only be accessed by authorized users. The authentication and authorization process is similar to the process when using a web browser.

1. Send the first HTTP request with the authentication credentials (e.g., user name and password) to the server. HTTP basic authentication is used.
   - If the authentication is successful, the session is created on the server and the JSESSIONID cookie is returned in the *Set-Cookie* response header.
2. Send another HTTP request, putting the JSESSIONID obtained in step one in the cookie request header. No credentials are needed in the request now.
   - If the HTTP response has a 401 status (unauthorized), the session created in step one does not exist anymore. A new authentication is needed. So, step one must be executed again.

⚠ In contrast to web browser usage, the consumer is responsible for getting and setting the JSSESSIONID cookie.

To access the repository API, the role ROLE_USER is needed. However, additional roles may be necessary depending on the requested functionality.

User role	Accessible repository API functionality
ROLE_REPOSITORY_READ	Operations returning read data in the response but not changing it on the server
ROLE_REPOSITORY_WRITE	Operations changing data on the server and returning the result in the response
ROLE_REPOSITORY_EVALUATE	Special writing operations executing Groovy-based DSL (= domain specific language) expressions on the server and returning the result in the response

Revision Differences

The MANTA Flow revision difference feature makes it possible to compare two different revisions. The differences are reported in a single CSV file.

Prerequisites

Authorization is required to use this feature. The user must have the roles `ROLE_USER` and `ROLE_EXPORTER`.

Usage

To use this feature, enter an HTTP GET request using the following format.

```
http://<server_name>:<port>/manta-dataflow-server/api/diff?oldRevision=<oldRevision>&newRevision=<newRevision>
```

where the `<server_name>` and `<port>` are provided by your application administrator.

The URL parameters:

Parameter	Description	Default value	Sample value
`<newRevision>`	The newer of the two revisions being compared	last committed revision	4
`<oldRevision>`	The older of the two revisions being compared	`<newRevision> - 1`	2

The HTTP response is of the `application/octet-stream` MIME type and contains the zipped difference report in CSV format.

Restrictions

- The last committed revision must be at least 1 (otherwise, there is nothing to compare).
- If both revisions are set, `<newRevision>` must be greater than `<oldRevision>`.
- If `<newRevision>` is set, it must be between 1 inclusive and the last committed revision inclusive.
- If `<oldRevision>` is set, it must be between 0 inclusive and the last committed revision exclusive.

Examples

Use case	HTTP GET request
Compare revisions 2 and 4	`http://<server_name>:<port>/manta-dataflow-server/api/diff?oldRevision=2&newRevision=4`

Compare revisions 3 and 4	`http://<server_name>:<port>/manta-dataflow-server/api/diff?oldRevision=3&newRevision=4`
	or
	`http://<server_name>:<port>/manta-dataflow-server/api/diff?newRevision=4`

Compare revision 2 and the last one committed	`http://<server_name>:<port>/manta-dataflow-server/api/diff?oldRevision=2`

Compare the last two revisions committed	`http://<server_name>:<port>/manta-dataflow-server/api/diff`

Usage with MANTA Flow Client

This feature is also available in MANTA Flow Client and the `manta-dataflow-revisiondiff-cli` module is required.

In MANTA Flow Client, it is only possible to compare the last two versions. To do this, run `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/revisionDifferenceScenario.bat` (on Windows) or `<MANTA_DIR_HOME>/scenarios/manta-dataflow-cli/bin/revisionDifferenceScenario.sh` (on Unix-like systems).

The difference report is stored in CSV format in the `<MANTA_DIR_HOME>/output/diff` folder.

Difference Report Format

The revision difference report is stored in a CSV file. The first line is a header; the following lines represent the corresponding differences. A difference record has the following structure.

Field group	Field	Description
Object changed between the compared revisions	Object resource	Resource of the changed object
	Object name	Qualified name of the changed object
	Object type	Type of the changed object
	Object status	The object change status; possible values: › ADDED—the object exists in the newer revision, not in the older one › REMOVED—the object exists in the older revision, not in the newer one › CHANGED—the object exists in both revisions, but its attributes are different or there is a difference in the connected objects
Connected object changed between the compared revisions This part of the record is empty if no connected object changed	Connected object resource	Resource of the connected object
	Connected object name	Qualified name of the connected object
	Connected object type	Type of the connected object
	Connected object status	The connected object status; possible values: › SOURCE_ADDED—the object is connected as a source in the newer version, not in the older one › SOURCE_REMOVED—the object is connected as a source in the older version, not in the newer one › TARGET_ADDED—the object is connected as a target in the newer version, not in the older one › TARGET_REMOVED—the object is connected as a target in the older version, not in the newer one

Example of a Report

diff.csv

```
"Object resource","Object name","Object type","Object status","Connected
object resource","Connected object name","Connected object type","
Connected object status"
"MSSQL","WIN-RRR5IO49B55/manta/dwh/SEGMENT_H","Table","ADDED","","","",""
"MSSQL","WIN-RRR5IO49B55/manta/dwh/SEGMENT_H/SEG_KEY","Column","
ADDED","","","",""
"MSSQL","WIN-RRR5IO49B55/manta/dwh/SEGMENT_H/VALID_TO","Column","
ADDED","","","",""
"MSSQL","WIN-RRR5IO49B55/manta/dwh/SEGMENT_H/VALID_FROM","Column","
```

```
ADDED","","","",""
"MSSQL","WIN-RRR5IO49B55/manta/dwh/SEGMENT_H/PARTY_KEY","Column","
ADDED","","","",""
"MSSQL","WIN-RRR5IO49B55/manta/dwh/HIST_PARTY","Procedure","ADDED","
MSSQL","WIN-RRR5IO49B55/manta/dwh/PARTY_H/BIRTH_DATE","Column","SOURCE
ADDED"
"MSSQL","WIN-RRR5IO49B55/manta/dwh/HIST_PARTY","Procedure","ADDED","
MSSQL","WIN-RRR5IO49B55/manta/dwh/PARTY_H/GENDER_KEY","Column","SOURCE
REMOVED"
"MSSQL","WIN-RRR5IO49B55/manta/dwh/HIST_PARTY","Procedure","ADDED","
MSSQL","WIN-RRR5IO49B55/manta/dwh/PARTY_H/GENDER_KEY","Column","TARGET
ADDED"
"MSSQL","WIN-RRR5IO49B55/manta/dwh/HIST_PARTY","Procedure","ADDED","
MSSQL","WIN-RRR5IO49B55/manta/dwh/PARTY_H/ICO","Column","TARGET ADDED"
"Oracle","ORCL/DWH/REP_CLIENT/ADDRESS_LINE2","Column","
REMOVED","","","",""
"Oracle","ORCL/DWH/REP_CLIENT/ADDRESS_LINE3","Column","
REMOVED","","","",""
```

The report says:

- A `SEGMENT_H` table with `SEG_KEY`, `VALID_TO`, `VALID_FROM`, and `PARTY_KEY` columns was added to a `dwh` schema (in a `manta` MS SQL database on the `WIN-RRR5IO49B55` server).
- The `BIRTH_DATE` column of the `dwh.PARTY_H` table became a source of the `HIST_PARTY` procedure.
- The `GENDER_KEY` column of the `dwh.PARTY_H` table stopped being a source of the `HIST_PARTY` procedure and became a target of the same procedure.
- The `ICO` column of the `dwh.PARTY_H` table stopped being a target of the `HIST_PARTY` procedure.
- The `ADDRESS_LINE2` and `ADDRESS_LINE3` columns were removed from the `REP_CLIENT` table or view in the `DWH` schema (in an `ORCL` Oracle database).