Informatica

Informatica® PowerExchange for Hive
10.5.7

User Guide

Informatica PowerExchange for Hive User Guide
10.5.7
December 2024

Publication Date: 2024-12-13

Table of Contents

Preface

Use the *Informatica® PowerExchange® for Hive User Guide* to learn how to read from or write to Hive by using the Developer tool. Learn to create a connection and develop mappings to access data in Hive sources and targets.

Informatica Resources

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

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Introduction to PowerExchange for Hive

This chapter includes the following topics:

PowerExchange for Hive Overview

Use PowerExchange for Hive to read from or write to Hive.

When you read from or write data to Hive, you run the mapping in the native or Hadoop environment. You can select the Blaze or Spark engine in the Hadoop environment to run the mapping.

You can run a mapping in the Hadoop environment when you want to optimize the performance to process large amounts of data. You can also configure PowerExchange for Hive to run on the Hadoop environment to read from or write to MapR-DB based Hive tables in JSON format.

When you run a mapping in the Hadoop environment, the Data Integration Service converts the mapping task to an execution plan based on the engine. It could be a Blaze execution plan or a Spark execution plan. In the Hadoop environment, the Data Integration Service converts the mapping to an execution plan that runs on a Hadoop cluster.

During mapping development, you can validate a Hive mapping for the native environment or the Hadoop environment.

For more information about configuring and running a mapping in the Hadoop environment, see the *Informatica Data Engineering Integration User Guide*.

Hive Data Extraction

Complete the following tasks to use PowerExchange for Hive to read data from Hive:

1. Create a Hive connection.
2. Import a relational data object.

3. Create a mapping and use the relational data object as a source to read data from Hive.

Hive Data Load

You can write to Hive only when you run mappings in the Hadoop environment. Use PowerExchange for Hive with Big Data Management to run mappings in the Hadoop environment.

Complete the following tasks to use PowerExchange for Hive to write data to Hive:

1. Create a Hive connection.
2. Import a relational data object.
3. Create a mapping and use the relational data object as a target to write data to Hive.
4. Specify the respective Hadoop run-time environment for the mapping and run the mapping to write data to Hive.

CHAPTER 2

PowerExchange for Hive Installation and Configuration

This chapter includes the following topic:

Prerequisites

PowerExchange for Hive requires services and environment variables to be available.

Before you can access Hive, perform the following tasks:

- Install and configure Informatica Services. Verify that the domain has a Data Integration Service and a Model Repository Service.
- Verify that a cluster configuration is created in the domain.
- Make sure that the HiveServer2 service is enabled on the Hadoop cluster.
- Verify that a Metadata Access Service is created in the domain.
- Verify that the Hadoop Distribution Directory property in the developerCore.ini file is set based on the Hadoop distribution that you use.
- When you run mappings on the Blaze engine, the metadata is fetched from the Hive table and data is read from and written to the underlying HDFS directory. Verify that the following privileges are granted:
  - Hadoop Impersonation user must have the SELECT privilege on Hive tables and required privilege for read or write on underlying HDFS directory.
  - Grant the Blaze user account CREATE TABLE permission on the Hive staging database specified in the Hadoop connection. The CREATE TABLE permission is required in the following situations:
    - The Hive source table uses SQL standard-based authorization.
    - A mapping contains a Lookup transformation with an SQL override.
  - When a mapping reads form a Hive source, verify that the mapping does not contain the columns that do not have access privileges.
- For an Update Strategy transformation that writes to a Hive target on the Blaze and Spark engines, verify that the user who runs the mapping has the SELECT privilege on all the columns of the target table along with the INSERT privilege on the target table.

CHAPTER 3

Hive Connections

This chapter includes the following topics:

Hive Connections Overview

After you configure PowerExchange for Hive, create a Hive connection.

You can use the Hive connection to access Hive as a source or target or to run mappings on a Hadoop cluster.

You can create a Hive connection using the Developer tool, Administrator tool, or Analyst tool.

Hive Connection Properties

Use the Hive connection to access Hive data. A Hive connection is a database type connection. You can create and manage a Hive connection in the Administrator tool, Analyst tool, or the Developer tool. Hive connection properties are case sensitive unless otherwise noted.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes Hive connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4000 characters.
Location	The domain where you want to create the connection. Not valid for the Analyst tool.
Type	The connection type. Select Hive.
LDAP username	LDAP user name of the user that the Data Integration Service impersonates to run mappings on a Hadoop cluster. The user name depends on the JDBC connection string that you specify in the Metadata Connection String or Data Access Connection String for the native environment. If the Hadoop cluster uses Kerberos authentication, the principal name for the JDBC connection string and the user name must be the same. Otherwise, the user name depends on the behavior of the JDBC driver. With Hive JDBC driver, you can specify a user name in many ways and the user name can become a part of the JDBC URL. If the Hadoop cluster does not use Kerberos authentication, the user name depends on the behavior of the JDBC driver. If you do not specify a user name, the Hadoop cluster authenticates jobs based on the following criteria: - The Hadoop cluster does not use Kerberos authentication. It authenticates jobs based on the operating system profile user name of the machine that runs the Data Integration Service. - The Hadoop cluster uses Kerberos authentication. It authenticates jobs based on the SPN of the Data Integration Service. LDAP username will be ignored.
Password	Password for the LDAP username.

Property	Description
Environment SQL	SQL commands to set the Hadoop environment. In native environment type, the Data Integration Service executes the environment SQL each time it creates a connection to a Hive metastore. If you use the Hive connection to run profiles on a Hadoop cluster, the Data Integration Service executes the environment SQL at the beginning of each Hive session. The following rules and guidelines apply to the usage of environment SQL in both connection modes: - Use the environment SQL to specify Hive queries. - Use the environment SQL to set the classpath for Hive user-defined functions and then use environment SQL or PreSQL to specify the Hive user-defined functions. You cannot use PreSQL in the data object properties to specify the classpath. If you use Hive user-defined functions, you must copy the .jar files to the following directory: `<Informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>/extras/hive-auxjars` - You can use environment SQL to define Hadoop or Hive parameters that you want to use in the PreSQL commands or in custom queries. - If you use multiple values for the Environment SQL property, ensure that there is no space between the values.
SQL Identifier Character	The type of character used to identify special characters and reserved SQL keywords, such as WHERE. The Data Integration Service places the selected character around special characters and reserved SQL keywords. The Data Integration Service also uses this character for the **Support mixed-case identifiers** property.

Properties to Access Hive as Source or Target

The following table describes the connection properties that you configure to access Hive as a source or target:

Property	Description
JDBC Driver Class Name	Name of the Hive JDBC driver class. If you leave this option blank, the Developer tool uses the default Apache Hive JDBC driver shipped with the distribution. If the default Apache Hive JDBC driver does not fit your requirements, you can override the Apache Hive JDBC driver with a third-party Hive JDBC driver by specifying the driver class name.
Metadata Connection String	The JDBC connection URI used to access the metadata from the Hadoop server. You can use PowerExchange for Hive to communicate with a HiveServer service or HiveServer2 service. To connect to HiveServer, specify the connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database name to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer2, use the connection string format that Apache Hive implements for that specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation. For user impersonation, you must add `hive.server2.proxy.user=<xyz>` to the JDBC connection URI. If you do not configure user impersonation, the current user's credentials are used connect to the HiveServer2. If the Hadoop cluster uses SSL or TLS authentication, you must add `ssl=true` to the JDBC connection URI. For example: `jdbc:hive2://<hostname>:<port>/<db>;ssl=true` If you use self-signed certificate for SSL or TLS authentication, ensure that the certificate file is available on the client machine and the Data Integration Service machine. For more information, see the*Data Engineering Integration Guide*.
Bypass Hive JDBC Server	JDBC driver mode. Select the check box to use the embedded JDBC driver mode. To use the JDBC embedded mode, perform the following tasks: - Verify that Hive client and Informatica services are installed on the same machine. - Configure the Hive connection properties to run mappings on a Hadoop cluster. If you choose the non-embedded mode, you must configure the Data Access Connection String. Informatica recommends that you use the JDBC embedded mode.
Fine Grained Authorization	When you select the option to observe fine grained authorization in a Hive source, the mapping observes the following: - Row and column level restrictions. Applies to Hadoop clusters where Sentry or Ranger security modes are enabled. - Data masking rules. Applies to masking rules set on columns containing sensitive data by Dynamic Data Masking. If you do not select the option, the Blaze and Spark engines ignore the restrictions and masking rules, and results include restricted or sensitive data.

Property	Description
Data Access Connection String	The connection string to access data from the Hadoop data store. To connect to HiveServer, specify the non-embedded JDBC mode connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer2, use the connection string format that Apache Hive implements for the specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation. For user impersonation, you must add `hive.server2.proxy.user=<xyz>` to the JDBC connection URI. If you do not configure user impersonation, the current user's credentials are used connect to the HiveServer2. If the Hadoop cluster uses SSL or TLS authentication, you must add `ssl=true` to the JDBC connection URI. For example: `jdbc:hive2://<hostname>:<port>/<db>;ssl=true` If you use self-signed certificate for SSL or TLS authentication, ensure that the certificate file is available on the client machine and the Data Integration Service machine. For more information, see the *Data Engineering Integration Guide*.
Hive Staging Directory on HDFS	HDFS directory for Hive staging tables. You must grant execute permission to the Hadoop impersonation user and the mapping impersonation users. This option is applicable and required when you write data to a Hive target in the native environment.
Hive Staging Database Name	Namespace for Hive staging tables. The Hive Staging Database Name is automatically updated from the Data Access Connection String. If you want to override the default name, you need to configure the Hive Staging Database Name in the Hive connection. This option is applicable when you run a mapping in the native environment to write data to a Hive target. If you run the mapping on the Blaze or Spark engine, you do not need to configure the Hive staging database name in the Hive connection. The Data Integration Service uses the value that you configure in the Hadoop connection.

Creating a Hive Connection

You must create a Hive connection to access Hive as a source or to run mappings on a Hadoop cluster.

Use the following procedure to create a Hive connection in the Developer tool:

1. Click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain in the **Available Connections**.
4. Select a connection type in **Database** > **Hive** and click **Add**.
5. Enter a connection name and optional description.
6. Click **Next**.

7. On the **Database Connection** page, you must choose the **Access Hive as a source or target** mode.

   Note: The **Use Hive to run mappings on a Hadoop cluster** mode is deprecated. To use the Hive driver to run mappings in the Hadoop cluster, use a Hadoop connection.
8. Enter the user name, password, and other attributes.
9. Select the cluster configuration associated with the Hadoop environment.
10. On the **Properties to Access Hive as Source and Target** page, you must specify the attributes as required.
11. If required, specify the properties to run mappings in the Hadoop cluster.
12. Click **Test Connection**. If a default Metadata Access Service is not set, a message appears to configure the Metadata Access Service. Click **OK** and set one Metadata Access Service as default. After you set a default Metadata Access Service, the connection to Hive is tested. If the Metadata Access Service does not exist, contact the Informatica administrator to create a new Metadata Access Service in the domain.
13. Click **Finish**.

CHAPTER 4

Data Objects for Hive

This chapter includes the following topics:

Data Objects for Hive Overview

Import a relational data object with a Hive connection to read data from or write to the Hive data warehouse.

After you import a relational data object, create a read or write transformation. Use the read or write transformation as a source or target in mappings and mapplets.

Relational Data Object

A relational data object is a physical data object that uses a relational table or view as a source.

Import a relational data object with a Hive connection to access data in the Hive data warehouse.

Create a relational data object to perform the following tasks:

- Filter rows when the Data Integration Service reads source data. If you include a filter condition, the Data Integration Service adds a `where` clause to the default query.
- Specify a join instead of the default join. If you include a user-defined join, the Data Integration Service replaces the join information specified by the metadata in the SQL query.
- Create a custom query to issue a special `SELECT` statement for the Data Integration Service to read source data. The custom query replaces the default query that the Data Integration Service uses to read data from sources.

You can include relational data objects in mappings and mapplets. You can add a relational data object to a mapping or mapplet as the following transformations:

- Read transformation if the run-time environment is native or Hadoop.
- Write transformation if the run-time environment is native or Hadoop.

Relational Data Object Properties

After you create a relational data object, you can modify the data object properties in the following data object views:

- **Overview** view. Use the **Overview** view to modify the relational data object name, description, and resources.
- **Advanced** view. Use the **Advanced** view to modify the run-time properties that the Data Integration Service uses.

When you add the relational data object to a mapping, you can edit the read or write properties.

Overview Properties

The **Overview** properties include general properties that apply to the relational data object. They also include column properties that apply to the resources in the relational data object.

General Properties

The following table describes the general properties that you configure for relational data objects:

Property	Description
Name	Name of the relational data object.
Description	Description of the relational data object.
Connection	Name of the relational connection.

Column Properties

The following table describes the column properties that you can view for relational data objects:

Property	Description
Name	Name of the column.
Native Type	Native data type of the column.
Precision	Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the column.

Advanced Properties

Advanced properties include run-time and other properties that apply to the relational data object.

The Developer tool displays advanced properties for relational data object in the **Advanced** view.

The following table describes the **Advanced** properties that you configure for a relational data object:

Property	Description
Connection	Name of the Hive connection.
Owner	Name of the Hive database.
Resource	Name of the resource.
Database Type	Type of the source. This property is read-only.
Resource Type	Type of the resource. This property is read-only.

Relational Data Object Read Properties

The data object operation properties include general, ports, query, sources, and advanced properties that the Data Integration Service uses to read data from Hive. Select the read transformation to edit the read properties.

Note: You cannot preview data from a Hive table that contains a partitioned column of Boolean data type.

General Properties

The general properties for the read transformation include the properties for name, description, and metadata synchronization.

The following table describes the general properties that you configure for the relational data object:

Property	Description
Name	Name of the relational data object. This property is read-only. You can edit the name in the **Overview** view. When you use the relational file as a source in a mapping, you can edit the name within the mapping.
Description	Description of the relational data object.
When column metadata changes	Indicates whether object metadata is synchronized with the source. Select one of the following options: - Synchronize output ports. The Developer tool reimports the object metadata from the source. - Do not synchronize. Object metadata may vary from the source.

Ports Properties

Ports properties include column names and column attributes such as data type and precision.

The following table describes the ports properties that you configure for relational sources:

Property	Description
Name	Name of the column.
Type	Native data type of the column.
Precision	Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the column.
Column	Name of the column in the source.
Resource	Name of the resource.

Query Properties

The Data Integration Service generates a default SQL query that it uses to read data from the relational resources. The default query is a SELECT statement for each column that it reads from the sources. You can override the default query through the simple or advanced query.

The following table describes the query properties that you configure for relational sources:

Property	Description
Show	Overrides the default query with a simple or advanced query. Use the simple query to select distinct values, enter a source filter, sort ports, or enter a user-defined join. Use the advanced query to create a custom SQL query for reading data from the sources.
Select Distinct	Selects unique values from the source. The Data Integration Service filters out unnecessary data when you use the relational data object in a mapping. **Note:** Select Distinct is disabled for columns with hierarchical data types.
Join	User-defined join in a relational data object. A user-defined join specifies the condition used to join data from multiple sources in the same relational data object.
Filter	Filter value in a read operation. The filter specifies the `where` clause of select statement. Use a filter to reduce the number of rows that the Data Integration Service reads from the source. When you enter a source filter, the Developer tool adds a `WHERE` clause to the default query.
Sort	Sorts the rows queried from the source. The Data Integration Service adds the ports to the `ORDER BY` clause in the default query. **Note:** You cannot use Sort for columns with hierarchical data types.
Advanced Query	Custom query. Use the advanced query to create a custom SQL query for reading data from the sources.

Run-time Properties

The run-time properties displays the name of the connection used for read transformation.

The following table describes the run-time properties that you configure for relational sources:

Property	Description
Connection	Name of the Hive connection.
Owner	Name of the Hive database.
Resource	Name of the resource.

Sources Properties

The sources properties lists the resources that are used in the relational data object and the source details for each of the resources. You can add or remove the resources. The Developer tool displays source properties for the read transformation.

Advanced Properties

The Data Integration Service runs the SQL commands when you use the relational data object in a mapping. The Data Integration Service runs pre-mapping SQL commands against the source database before it reads the source. It runs post-mapping SQL commands against the source database after it writes to the target.

The file path in an SQL command depends on the type of the run-time environment. If you run the mapping in the native environment, the file path must be relative to the host that you specified in the Hive connection. If you run the mapping in the Hadoop environment, the file path must be relative to the machine that hosts the Data Integration Service for the Hadoop environment type.

The following table describes the advanced properties that you configure for Hive sources:

Property	Description
Tracing level	Controls the amount of detail in the mapping log file.
PreSQL	SQL command the Data Integration Service runs against the source database before it reads the source. The Data Integration Service and the Spark engine can run PreSQL commands against Hive sources. The Developer tool does not validate the SQL.
PostSQL	SQL command the Data Integration Service runs against the source database after it writes to the target. The Data Integration Service and the Spark engine can run PostSQL commands against Hive sources. The Developer tool does not validate the SQL.

Relational Data Object Write Properties

The data object operation properties include general, ports, run-time, target, and advanced properties that the Data Integration Service uses to write data to Hive. Select the write transformation to edit the write properties.

Target Schema Strategy

When you write data to a Hive target, you can configure a target schema strategy. Click the **Advanced** tab configure the target schema strategy.

You can select one of the following target schema strategies:

RETAIN - Retain existing target schema

The Data Integration Service retains the existing target schema.

CREATE - Create or replace table at run time

The Data Integration Service drops the target table at run time and replaces it with a table based on a target table that you identify.

APPLYNEWCOLUMNS - Alter table and apply new columns only

The Data Integration Service alters the target table by applying new columns from the associated data object or mapping flow to the target table. If the mapping flow includes changes to existing columns in the target table, the Data Integration Service does not apply these changes to the target table. This option is applicable when you run the mapping in the native environment or on the Spark engine.

APPLYNEWSCHEMA - Alter table and apply new schema

The Data Integration Service alters the target table and applies the new schema from the associated data object or mapping flow to the target table. The Data Integration Service applies changes to existing columns and adds new columns to the target table based on the mapping flow. This option is applicable when you run the mapping in the native environment or on the Spark engine.

Note:

- If you apply a new schema to a partitioned column in a Hive target, the mapping fails. You cannot drop or rename columns in a Hive target by using the APPLYNEWSCHEMA target schema strategy.
- If a column is missing in the associated data object or mapping flow, then null values are inserted into the table.

FAIL - Fail mapping if target schema is different

The Data Integration Service fails the mapping if the target schema from the mapping flow is different from the schema of the target table. This option is applicable when you run the mapping in the native environment or on the Spark engine.

Assign Parameter

You can assign a parameter to represent the value for the target schema strategy and then change the parameter at run time.

General Properties

The general properties for the write transformation include the properties for name, description, and metadata synchronization.

The following table describes the general properties that you configure for the relational data object:

Property	Description
Name	Name of the relational data object. This property is read-only. You can edit the name in the **Overview** view. When you use the relational file as a source in a mapping, you can edit the name within the mapping.
Description	Description of the relational data object.
When column metadata changes	Indicates whether object metadata is synchronized with the source. Select one of the following options: - Synchronize output ports. The Developer tool reimports the object metadata from the source. - Do not synchronize. Object metadata may vary from the source.

Ports Properties

Ports properties include column names and column attributes such as data type and precision.

The following table describes the ports properties that you configure for relational targets:

Property	Description
Name	Name of the column.
Type	Native data type of the column.
Precision	Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the column.
Column	Name of the column in the resource.
Resource	Name of the resource.

Run-time Properties

The run-time properties displays the connection name and reject file and directory.

The following table describes the run-time properties that you configure for relational targets:

Property	Description
Connection	Name of the Hive connection.
Owner	Name of the Hive database.
Resource	Name of the resource.
Reject truncated/ overflows rows	Write truncated and overflow data to the reject file. If you select Reject Truncated/Overflow Rows, the Data Integration Service sends all truncated rows and any overflow rows to the reject file. This property is not applicable to Hive targets.
Reject file directory	Directory where the reject file exists. This property is not applicable to Hive targets.
Reject file name	File name of the reject file. This property is not applicable to Hive targets.

Target Properties

The target properties lists the resource that is used in the relational data object and the target details for the resource. The Developer tool displays target properties for the write transformation.

Advanced Properties

The advanced properties includes the write properties used to write data to the target. You can specify properties such as SQL commands.

The file path in an SQL command depends on the type of the run-time environment. If you run the mapping in the native environment, the file path must be relative to the host that you specified in the Hive connection. If you run the mapping in the Hadoop environment, the file path must be relative to the machine that hosts the Data Integration Service for the Hive environment type.

The following table describes the advanced properties that you configure for Hive targets:

Property	Description
Tracing level	Controls the amount of detail in the mapping log file.
Target Schema Strategy	Type of target schema strategy for the target table. You can select one of the following target schema strategies: - RETAIN - Retain existing target schema - CREATE - Create or replace table at run time - APPLYNEWCOLUMNS - Alter table and apply new columns only - APPLYNEWSCHEMA - Alter table and apply new schema - FAIL - Fail mapping if target schema is different - Assign Parameter

Property	Description
DDL query for create or replace	The DDL query based on which the Data Integration Service creates or replaces the target table. This option is applicable when you select the **CREATE - Create or replace table at run time** target schema strategy. By default, a Hive table is created with ORC storage format on the Hortonworks HDP 3.1 distribution.
Truncate target table	Truncates the target before loading data. Default is enabled.
Truncate target partition	Truncates an internal or external partitioned Hive target before loading data. You must select the **Truncate target table** option before you select this option. This option is applicable when you run the mapping in the Hadoop environment. Default is disabled.
PreSQL	SQL command that the Data Integration Service runs against the target database before it reads the source. Only the Spark engine can run PreSQL commands against Hive targets. The Developer tool does not validate the SQL.
PostSQL	SQL command that the Data Integration Service runs against the target database after it writes to the target. Only the Spark engine can run PostSQL commands against Hive targets. The Developer tool does not validate the SQL.
Maintain row order	Maintains row order while writing data to the target. Select this option if the Data Integration Service should not perform any optimization that can change the row order. When the Data Integration Service performs optimizations, it might lose the row order that was established earlier in the mapping. When you configure a target to maintain row order, the Data Integration Service does not perform optimizations for the target.

Importing a Relational Data Object with a Hive Connection

Import a relational data object with a Hive connection to access data in the Hive data warehouse.

Before you import a relational data object, you configure a Hive connection.

1. Select a project or folder in the Object Explorer view.
2. Click **File** > **New** > **Data Object**.
3. Select **Relational Data Object** and click **Next**.

   The **New Relational Data Object** dialog box appears.
4. Click **Browse** next to the Connection option and select the Hive connection from which you want to import the Hive resources.

5. Click **Create data from existing resource**.
6. To add a resource to the Relational Data Object, click **Browse** next to the Resource option.

   If a default Metadata Access Service is not set, a message appears to configure the Metadata Access Service. Click **OK** and set one Metadata Access Service as default. After you set a default Metadata Access Service, the **Add sources to the data object** dialog box appears. If the Metadata Access Service does not exist, contact the Informatica administrator to create a new Metadata Access Service in the domain.
7. Select a table from a schema to add to the data object.

   The **Show Default Schema Only** option is selected by default. If you want to specify multiple schema names, you must clear the **Show Default Schema Only** option to view the tables under the specified schema names.
8. Navigate to the table to add it to the data object and click **OK**.
9. Click **Browse** next to the Location option and select the target project or folder.
10. Click **Finish**.

    The data object appears under Data Object in the project or folder in the **Object Explorer** view. You can also add resources to a relational data object after you create it.

Troubleshooting the Relational Data Object Import

The solution to the following situation might help you troubleshoot the relational data object import task:

I see SocketTimeOutException while importing a Hive table.

The default socket timeout is set to 60 seconds. Perform the following steps to increase the socket timeout:

1. Open the `developerCore.ini` file located at `<INFA_CLIENT_HOME>\DeveloperClient\`.
2. Append the following line of code: `-Dlogin.timeout=<socket timeout in seconds>`. For example: `-Dlogin.timeout=120`
3. Save the `developerCore.ini` file.
4. Run the following command from the command prompt: `<INFA_CLIENT_HOME>\DeveloperClient\developer.exe -clean`

Creating a Read or Write Transformation

Create a read or write transformation to add it to a mapping or mapplet.

1. Open the mapping or mapplet in which you want to create a read or write transformation.
2. In the **Object Explorer** view, select one or more relational data objects.
3. Drag the relational data objects into the mapping editor.

   The **Add to Mapping** dialog box appears.
4. Select the **Read** or **Write** based on the environment type.
   - Select **Read** if the validation or run-time environment is native or Hadoop.
   - Select **Write** if the validation or run-time environment is native or Hadoop.

5. Click **OK.**

   The Developer tool creates a read or write transformation for the relational data object in the mapping or mapplet.

CHAPTER 5

Hive Mappings

This chapter includes the following topics:

Hive Mappings Overview

After you create the relational data object with a Hive connection, you can develop a mapping. You can define the following types of objects in the mapping:

- A read transformation of the relational data object to read data from Hive in native or Hadoop run-time environment.
- A target or a write transformation of the relational data object to write data to Hive in native or Hadoop run-time environment.

Validate and run the mapping. You can deploy the mapping and run it or add the mapping to a Mapping task in a workflow.

You can create a Lookup transformation from Hive objects in mappings in the native environment. However, you cannot use the dynamic lookup cache. You cannot create an uncached lookup and a lookup on Logical Data Objects.

Mapping Validation and Run-time Environments

You can validate and run mappings in the native environment or a Hadoop environment.

You can validate a mapping to run in the native environment, Hadoop environment, or both. The Data Integration Service validates whether the mapping can run in the selected environment. You must validate the mapping for an environment before you run the mapping in that environment.

When you run a mapping in the native environment, the Data Integration Service runs the mapping from the Developer tool.

When you run a mapping in the Hadoop environment, the Data Integration Service converts the mapping task to an execution plan based on the engine. It could be a Blaze execution plan or Spark execution plan. In the Hadoop environment, the Data Integration Service converts the mapping to an execution plan that is run on a Hadoop cluster.

Audits

To validate the consistency and accuracy of data processed in a mapping for a read or write operation, you can create an audit for the mapping.

An audit is composed of rules and conditions. Use a rule to compute an aggregated value for a single column of data. Use a condition to make comparisons between multiple rules or between a rule and constant values.

You can run audits with mappings that run on the Data Integration Service or the Spark engine.

For more information, see the *Data Engineering Integration 10.5 User Guide*.

Hive Mapping Example

Your organization, HypoMarket Corporation, needs to analyze customer data. Create a mapping that reads all the customer records. Create an SQL data service to make a virtual database available for end users to query.

You can use the following objects in a Hive mapping:

Hive input

The input file is a Hive table that contains the customer names and contact details.

Create a relational data object. Configure the Hive connection and specify the table that contains the customer data as a resource for the data object. Drag the data object into a mapping as a read data object.

SQL Data Service output

Create an SQL data service in the Developer tool. To make it available to end users, include it in an application, and deploy the application to a Data Integration Service. When the application is running, connect to the SQL data service from a third-party client tool by supplying a connect string.

You can run SQL queries through the client tool to access the customer data.

Rules and Guidelines for Hive mappings

Consider the following rules and guidelines for Hive mappings:

- When you alter a Hive table by adding a new column to a Hive partition table and run a mapping in HDP 2.6 distribution, the mapping fails.
- If the Hive object in a mapping configured for an Insert operation is set to false and you run the mapping on the Blaze engine or the native environment, the mapping fails on Cloudera CDH 6.1. You must create a new source and target physical data object and run the mapping.

APPENDIX A

Data Type Reference

This appendix includes the following topics:

Data Type Reference Overview

Informatica Developer uses the following data types in Hive mappings:

Hive native data types

Hive native data types appear in the physical data object column properties.

Transformation data types

Transformation data types are set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Data Integration Service uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

Transformation data types include the following data types:

- Primitive data type. Represents a single data value in a single column position.
- Complex data type. Represents multiple data values in a single column position. Use complex data types in mappings that run on the Spark engine to process hierarchical data in complex files.

When the Data Integration Service reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the Data Integration Service writes to a target, it converts the transformation data types to the comparable native data types.

Hive Complex Data Types

Hive complex data types such as arrays, maps, and structs are a composite of primitive or complex data types. Informatica Developer represents complex data types with the string data type and uses delimiters to separate the elements of the complex data type.

You can read and write hierarchical data from Hive tables in a mapping that runs on the Spark engine.

The following table lists the complex data types:

Complex Data Type	Description
Array	An array contains an ordered collection of elements of same data type. The elements in an array are delimited by commas. For example, an array of `fruits` is represented as `[apple,banana,orange]`.
Map	A map contains an unordered collection of key-value pairs and are represented as pairs of strings and integers delimited by the = character. String and integer pairs are delimited by commas. For example, a map of `fruits` is represented as `[1=apple,2=banana,3=orange]`.
Struct	A struct contains a collection of elements of different data types delimited by the : character. String and integer pairs are delimited by commas. For example, a struct is represented as `struct {1:"apple" [, "apple":"red", ...]} .`

Hive Data Types and Transformation Data Types

The following table lists the Hive data types that Data Integration Service supports and the corresponding transformation data types:

Hive Data Type	Transformation Data Type	Range and Description
Binary	Binary	1 to 104,857,600 bytes. You can read and write data of Binary data type in a Hadoop environment. You can use the user-defined functions to transform the binary data type.
Tiny Int	Integer	-32,768 to 32,767
Integer	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Bigint	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
Decimal	Decimal	For transformations that support precision up to 38 digits, the precision is 1 to 38 digits, and the scale is 0 to 38. For transformations that support precision up to 28 digits, the precision is 1 to 28 digits, and the scale is 0 to 28. If a mapping is not enabled for high precision, the Data Integration Service converts all decimal values to double values. If a mapping is enabled for high precision, the Data Integration Service converts decimal values with precision greater than 38 digits to double values.
Double	Double	Precision 15

Hive Data Type	Transformation Data Type	Range and Description
Float	Double	Precision 15
String	String	1 to 104,857,600 characters
Boolean	Integer	TRUE (1) or FALSE (0). The default transformation type for boolean is integer. You can also set this to string data type with values of True and False.
Array	Array	Unlimited number of characters.
Struct	Struct	Unlimited number of characters.
Map	Map	Unlimited number of characters.
Timestamp	datetime	January 1, 0001 00:00:00 to December 31, 9999 23:59:59.997. Precision to microsecond.
Date	datetime	January 1, 0001 to December 31, 9999.
Char	String	1 to 255 characters
Varchar	String	1 to 65355 characters

INDEX