Informatica® PowerExchange for Hive
10.1.1 Update 2

User Guide

Informatica PowerExchange for Hive User Guide
10.1.1 Update 2
March 2017

Publication Date: 2018-09-27

Table of Contents

Preface

The *Informatica PowerExchange® for Hive User Guide* provides information about how to use PowerExchange for Hive with Informatica Data Services and Informatica Data Explorer to access data in Hive and move that information to targets for analysis. The guide is written for database administrators and developers who are responsible for developing mappings that access data in Hive sources.

This guide assumes that you have knowledge of Hive and Informatica.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- 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 search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.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

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

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

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access 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. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access 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 augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

Introduction to PowerExchange for Hive

This chapter includes the following topics:

PowerExchange for Hive Overview

Use PowerExchange for Hive to read data from a Hive source. Use PowerExchange for Hive with Big Data Management™ to write data to a Hive target.

When you write data to a Hive target, you run the mapping in a Hive environment. In a Hive environment, the Data Integration Service converts the mapping to a Hive execution plan that is run in a Hadoop cluster.

When you read data from Hive, you can run the mapping in the native environment. In the native environment, the DIS runs the mapping from the Developer tool. You can optionally run a mapping with a Hive source in a Hive environment when you have Big Data Management. You might run a mapping in a Hive environment to optimize performance when you need to process large amounts of data.

During mapping development, you validate a Hive mapping for the native environment or a Hive environment. You use a Hive connection to connect to Hive to read and write Hive data. You can also use a Hive connection to specify the Hive validation and run-time environments.

For more information about configuring and running a mapping in a Hive environment, see the *Big Data Management 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 Hive environment. Use PowerExchange for Hive with Big Data Management to run mappings in the Hive 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 Hive 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:

1. Install and configure Informatica Data Services.
2. Install a Thrift server.
3. Start the Hive server as a Thrift service.
4. If you are using a remote metastore for the Hive run-time environment, start the Hive Metastore Service. For information about the Thrift server, see http://thrift.apache.org/.
5. On the Developer tool machine, verify that the Hadoop Distribution directory property is set in the developer.ini file.
   The default value of the property is `-DINFA_HADOOP_DIST_DIR=hadoop\cloudera_cdh3u4`
6. To use a Hive connection to run mappings in the Hadoop cluster, verify that the following properties are set on the Processes tab of the Data Integration Service page in the Administrator tool:
   - Data Integration Service Hadoop Distribution directory
   - Hadoop Distribution directory
7. For Blaze mode, when a mapping runs, the metadata is fetched from the Hive table and data is read from and written to the underlying HDFS directory. Therefore, Hadoop Impersonation user must have the SELECT privilege on Hive tables and required privilege for read or write on underlying HDFS directory. Column-level privileges are not considered for read and write operations for Hive tables.

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 in the Hadoop cluster.

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

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.
Connection Modes	Hive connection mode. Select at least one of the following options: - Access Hive as a source or target. Select this option if you want to use Hive as a source or a target. - Use Hive to run mappings in Hadoop cluster. Select this option if you want to use the Hive driver to run mappings in the Hadoop cluster.

Property	Description
User Name	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 runs Hortonworks HDP, you must provide a user name. If you use Tez to run mappings, you must provide the user account for the Data Integration Service. If you do not use Tez to run mappings, you can use an impersonation user account. 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.
Common Attributes to Both the Modes: 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 in the 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. The path must be the fully qualified path to the JAR files used for user-defined functions. Set the parameter hive.aux.jars.path with all the entries in infapdo.aux.jars.path and the path to the JAR files for user-defined functions. - 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. The following sample text shows two values that can be used for the Environment SQL: `set hive.execution.engine='tez';set hive.exec.dynamic.partition.mode='nonstrict';` If you use the Hive connection to run profiles in the Hadoop cluster, the Data Integration service executes only the environment SQL of the Hive connection. If the Hive sources and targets are on different clusters, the Data Integration Service does not execute the different environment SQL commands for the connections of the Hive source or target.

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
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 HiveServer 2, 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. 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 *Big Data Management Installation and Configuration 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 in the 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.
Observe Fine Grained SQL Authorization	When you select the option to observe fine-grained SQL authentication in a Hive source, the mapping observes row and column-level restrictions on data access. If you do not select the option, the Blaze run-time engine ignores the restrictions, and results include restricted data.
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 HiveServer 2, 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. 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 *Big Data Management Installation and Configuration Guide*.

Properties to Run Mappings in Hadoop Cluster

The following table describes the Hive connection properties that you configure when you want to use the Hive connection to run Informatica mappings in the Hadoop cluster:

Property	Description
Database Name	Namespace for tables. Use the name `default` for tables that do not have a specified database name.
Default FS URI	The URI to access the default Hadoop Distributed File System. Use the following connection URI: `hdfs://<node name>:<port>` Where - <node name> is the host name or IP address of the NameNode. - <port> is the port on which the NameNode listens for remote procedure calls (RPC). If the Hadoop cluster runs MapR, use the following URI to access the MapR File system: `maprfs:///`.
JobTracker/Yarn Resource Manager URI	The service within Hadoop that submits the MapReduce tasks to specific nodes in the cluster. Use the following format: `<hostname>:<port>` Where - <hostname> is the host name or IP address of the JobTracker or Yarn resource manager. - <port> is the port on which the JobTracker or Yarn resource manager listens for remote procedure calls (RPC). If the cluster uses MapR with YARN, use the value specified in the `yarn.resourcemanager.address` property in yarn-site.xml. You can find `yarn-site.xml` in the following directory on the NameNode of the cluster: `/opt/mapr/hadoop/hadoop-2.5.1/etc/hadoop`. MapR with MapReduce 1 supports a highly available JobTracker. If you are using MapR distribution, define the JobTracker URI in the following format: `maprfs:///`
Hive Warehouse Directory on HDFS	The absolute HDFS file path of the default database for the warehouse that is local to the cluster. For example, the following file path specifies a local warehouse: `/user/hive/warehouse` For Cloudera CDH, if the Metastore Execution Mode is remote, then the file path must match the file path specified by the Hive Metastore Service on the Hadoop cluster. For MapR, use the value specified for the `hive.metastore.warehouse.dir` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node that runs HiveServer2: `/opt/mapr/hive/hive-0.13/conf`.

Property	Description

Advanced Hive/Hadoop Properties	Configures or overrides Hive or Hadoop cluster properties in hive-site.xml on the machine on which the Data Integration Service runs. You can specify multiple properties.
	Select **Edit** to specify the name and value for the property. The property appears in the following format:
	`<property1>=<value>`
	Where
	- <property1> is a Hive or Hadoop property in hive-site.xml.
	- <value> is the value of the Hive or Hadoop property.
	When you specify multiple properties, `&:` appears as the property separator.
	The maximum length for the format is 1 MB.
	If you enter a required property for a Hive connection, it overrides the property that you configure in the Advanced Hive/Hadoop Properties.
	The Data Integration Service adds or sets these properties for each map-reduce job. You can verify these properties in the JobConf of each mapper and reducer job. Access the JobConf of each job from the Jobtracker URL under each map-reduce job.
	The Data Integration Service writes messages for these properties to the Data Integration Service logs. The Data Integration Service must have the log tracing level set to log each row or have the log tracing level set to verbose initialization tracing.
	For example, specify the following properties to control and limit the number of reducers to run a mapping job:
	`mapred.reduce.tasks=2&:hive.exec.reducers.max=10`

Temporary Table Compression Codec	Hadoop compression library for a compression codec class name.
Codec Class Name	Codec class name that enables data compression and improves performance on temporary staging tables.
Metastore Execution Mode	Controls whether to connect to a remote metastore or a local metastore. By default, local is selected. For a local metastore, you must specify the Metastore Database URI, Driver, Username, and Password. For a remote metastore, you must specify only the Remote Metastore URI.

Metastore Database URI	The JDBC connection URI used to access the data store in a local metastore setup. Use the following connection URI:
	`jdbc:<datastore type>://<node name>:<port>/<database name>`
	where
	- <node name> is the host name or IP address of the data store.
	- <data store type> is the type of the data store.
	- <port> is the port on which the data store listens for remote procedure calls (RPC).
	- <database name> is the name of the database.
	For example, the following URI specifies a local metastore that uses MySQL as a data store:
	`jdbc:mysql://hostname23:3306/metastore`
	For MapR, use the value specified for the `javax.jdo.option.ConnectionURL` property in `hive-site.xml`. You can find hive-site.xml in the following directory on the node where HiveServer 2 runs: /opt/mapr/hive/hive-0.13/conf.

Property	Description
Metastore Database Driver	Driver class name for the JDBC data store. For example, the following class name specifies a MySQL driver: `com.mysql.jdbc.Driver` For MapR, use the value specified for the `javax.jdo.option.ConnectionDriverName` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.
Metastore Database Username	The metastore database user name. For MapR, use the value specified for the `javax.jdo.option.ConnectionUserName` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.
Metastore Database Password	The password for the metastore user name. For MapR, use the value specified for the `javax.jdo.option.ConnectionPassword` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.
Remote Metastore URI	The metastore URI used to access metadata in a remote metastore setup. For a remote metastore, you must specify the Thrift server details. Use the following connection URI: `thrift://<hostname>:<port>` Where - <hostname> is name or IP address of the Thrift metastore server. - <port> is the port on which the Thrift server is listening. For MapR, use the value specified for the `hive.metastore.uris` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.

Creating a Hive Connection

You must create a Hive connection to access Hive as a source or to run mappings in the 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 **Connection Details** page, you must choose the Hive connection mode and specify the commands for environment SQL. The SQL commands are applicable to both the connection modes. Select at least one of the following connection modes:

- Access Hive as a source or target. Use the connection to access Hive data. If you select this option, the **Properties to Access Hive as a source or target** page appears. Configure the connection strings.
- Use Hive to run mappings in the Hadoop cluster. Use the Hive connection to validate and run Informatica mappings in the Hadoop cluster. If you select this option, the **Properties used to Run Mappings in the Hadoop Cluster** page appears. Configure the properties.

8. Click **Test Connection** to verify the Hive connection.

   You can test a Hive connection that is configured to access Hive data. You cannot test a Hive connection that is configured to run Informatica mappings in a Hive environment.

9. 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 Hive.
- Write transformation if the run-time environment is Hive.

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.
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.
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 Hive 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 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 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 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.

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 Hive 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.
PreSQL	SQL command that the Data Integration Service runs against the target database before it reads the source. The Developer tool does not validate the SQL. This property is not applicable to Hive targets.

Property	Description
PostSQL	SQL command that the Data Integration Service runs against the target database after it writes to the target. The Developer tool does not validate the SQL. This property is not applicable to Hive targets.
Truncate target table	Truncates the target before loading data. **Note:** If the mapping target is a Hive partition table, you can choose to truncate the target table only with Hive version 0.11.

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 Location option and select the target project or folder.
5. Click **Browse** next to the Connection option and select the Hive connection from which you want to import the Hive resources.
6. To add a resource to the Relational Data Object, click **Add** next to the Resource option.

   The **Add sources to the data object** dialog box appears.
7. Navigate to the resource to add it to the data object and click **Ok**.
8. 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 Hive.
   - Select **Write** if the validation or run-time environment is Hive.
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 Hive run-time environment.
- Transformations.
- A target or a write transformation of the relational data object to write data to Hive only if the run-time environment is Hive.

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.

Hive Validation and Run-time Environments

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

You can validate a mapping to run in the native environment, Hive 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 a Hive environment, the Data Integration Service converts the task into HiveQL queries to enable the Hadoop cluster to process the data.

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.

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 data types appear in the physical data object column properties.
- Transformation data types. 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.

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.

Note: Hive complex data types in a Hive source or Hive target are not supported when you run mappings in a Hadoop cluster.

The following table describes the transformation types and delimiters that are used to represent the complex data types:

Complex Data Type	Description
Array	The elements in the array are of string data type. The elements in the array are delimited by commas. For example, an array of `fruits` is represented as `[apple,banana,orange]`.
Map	Maps contain 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	Structs are represented as pairs of strings and integers delimited by the : character. String and integer pairs are delimited by commas. For example, a struct of `fruits` is represented as `[1,apple]`.

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

Hive Data Type	Transformation Data Type	Range and Description
Decimal	Decimal	Precision 1 to 28, scale 0 to 28 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. 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
Float	Double	Precision 15
String	String	1 to 104,857,600 characters
Boolean	Integer	1 or 0 The default transformation type for boolean is integer. You can also set this to string data type with values of True and False.
Arrays	String	1 to 104,857,600 characters
Struct	String	1 to 104,857,600 characters
Maps	String	1 to 104,857,600 characters
Timestamp	datetime	The time stamp format is YYYY-MM-DD HH:MM:SS.fffffffff. Precision 29, scale 9.
Date	datetime	0000-0101 to 999912-31. Hive date format is YYYY-MM-DD. Precision 10, scale 0.
Char	String	1 to 255 characters
Varchar	String	1 to 65355 characters

INDEX

D

H

M

P