Informatica

Informatica® PowerExchange for Kudu 10.5.6

User Guide

Informatica PowerExchange for Kudu User Guide
10.5.6
May 2024

Publication Date: 2024-05-30

Table of Contents

Preface

Use the *Informatica® PowerExchange® for Kudu User Guide* to learn how to write to Kudu by using the Developer tool. Learn to create a connection, develop and run mappings and dynamic mappings on the Spark engine in the Hadoop environment.

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 Kudu

This chapter includes the following topics:

PowerExchange for Kudu Overview

You can use PowerExchange for Kudu to connect to Kudu from Informatica.

Use PowerExchange for Kudu to write data to Kudu. You can use Kudu objects as targets in mappings and dynamic mappings. You can validate and run mappings on the Spark engine in the Hadoop environment.

Introduction to Kudu

Kudu is a columnar storage manager developed for the Apache Hadoop platform. You can use Kudu to store data in tables. Kudu has a simple data model, where the Kudu table has a primary key made up of one or more columns, each with a defined type. Kudu tables have a columnar structure which helps to vectorize and compress data easily.

Use Kudu to perform real-time analytics on fast data. You can use Kudu for fast data search, updates and inserts.

CHAPTER 2

PowerExchange for Kudu Configuration

This chapter includes the following topics:

PowerExchange for Kudu Configuration Overview

PowerExchange for Kudu installs with the Informatica services and clients.

To configure PowerExchange for Kudu, complete the prerequisites.

Prerequisites

Before you use PowerExchange for Kudu, you must complete the following prerequisites:

- Install and configure the Informatica services.
- Install and configure the Developer tool. You can install the Developer tool when you install Informatica clients.
- Create a Data Integration Service and a Model Repository Service in the Informatica domain.
- Verify that the user used to configure the Informatica domain is added to the cluster and the user has `sudo` privileges to use non-kerberised Hadoop distribution.
- 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.

Java Heap Memory Configuration (Optional)

Configure the memory for the Java heap size in the node that runs the Data Integration Service.

1. In the Administrator tool, navigate to the Data Integration Service for which you want to change the Java heap size.
2. Edit the Custom Properties section in the Data Integration Service Properties.
3. To increase the heap memory size for a large dataset, define the following properties:

   ```
   ExecutionContextOptions.JVMMaxMemory = <size> MB
   ExecutionContextOptions.JVMMinMemory = <size> MB
   ```

   Where <size> is a valid heap size, such as 2048 MB.
4. Click **OK**.
5. Restart the Data Integration Service.

CHAPTER 3

PowerExchange for Kudu Connections

This chapter includes the following topics:

PowerExchange for Kudu Connection Overview

You can use PowerExchange for Kudu connection to write data to Kudu.

You can use a PowerExchange for Kudu connection to create data objects and run mappings. The Developer tool uses the connection when you create a data object. The Data Integration Service uses the connection when you run mappings.

You can create a PowerExchange for Kudu connection from the Developer tool or the Administrator tool. Create and manage connections in the Informatica Preferences window or the Connection Explorer view.

Kudu Connection Properties

Use a Kudu connection to access Kudu.

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

You can create and manage a Kudu connection in the Administrator tool or the Developer tool. The following table describes the Kudu 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 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select Kudu.

The following table describes the properties for metadata access:

Property	Description
Kudu Master URLs	The URLs of the Kudu master tables.
Kudu Library Version	The version number of the Kudu library.
Cluster Configuration	The Hadoop cluster that you use for the connection.

CHAPTER 4

PowerExchange for Kudu Data Objects

This chapter includes the following topics:

PowerExchange for Kudu Data Object Overview

A Kudu data object is a physical data object that represents a Kudu table as a target. A Kudu data object is the representation of data that is based on a Kudu object.

You can configure the data object write operation properties that determine how data can be loaded to Kudu.

To write data to the Kudu, create a data object write operation based on the Kudu data object. Configure the write operation properties to determine how the Data Integration Service must write data to the Kudu. Add the write operation as a Write transformation in a mapping.

PowerExchange for Kudu Data Object Write Operation

Create a mapping to write data to Kudu. Use the Kudu connection, and define the write operation properties to write data to Kudu.

PowerExchange for Kudu Data Object Write Operation Properties

The Kudu data object write operation properties include run-time properties that apply to the Kudu data object.

The Developer tool displays advanced properties for the Kudu data object operation in the Advanced view.

The following table describes the Advanced properties for a Kudu data object write operation:

Property	Description
Number of Replica	Sets the number of replicas that each table has. You must set the replication factor to an odd number. Default is 1.
Table Name	Name of the target table.
Table Partitioning Strategy	Type of partition applicable to the table. You can select from the following options: - HASH: In hash user keys partitioning, the Integration Service uses a hash function to group rows of data among partitions based on a user-defined partition key. - RANGE: With range partitioning, the Integration Service distributes rows of data based on a port or set of ports that you define as the partition key. Partitioning is applicable to only the primary key fields. For create target Hash default is 3 bucket partition and Range default is 1 range partition and you cannot specify the range conditions for the columns of your choice. Default is HASH.
Upsert	Upserts data into a Kudu target. To perform an upsert operation, you must add an Update Strategy transformation to a mapping before the data object write operation and flag the records for upsert in an update strategy expression. Otherwise, the Data Integration Service performs an insert operation on the Kudu target.
Target Schema Strategy	Target schema strategy for the Kudu 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 When you select the CREATE option, you must provide the value of the Schema Name property to run the mapping successfully. - Assign Parameter.

Creating a PowerExchange for Kudu Data Object Operation

You can create a data object write operation for Kudu data objects and add the Kudu data object operation to a mapping.

1. Select the data object in the **Object Explorer** view.

2. Right-click and select **New** > **Data Object Operation**.

   The **Data Object Operation** dialog box appears.
3. Enter a name for the data object operation.
4. Select the type of data object operation. You can choose to create a write operation.
5. Click **Add**.

   The **Select Resources** dialog box appears.
6. Select the Kudu data object for which you want to create the data object operation and click **OK**.
7. Click **Finish**.

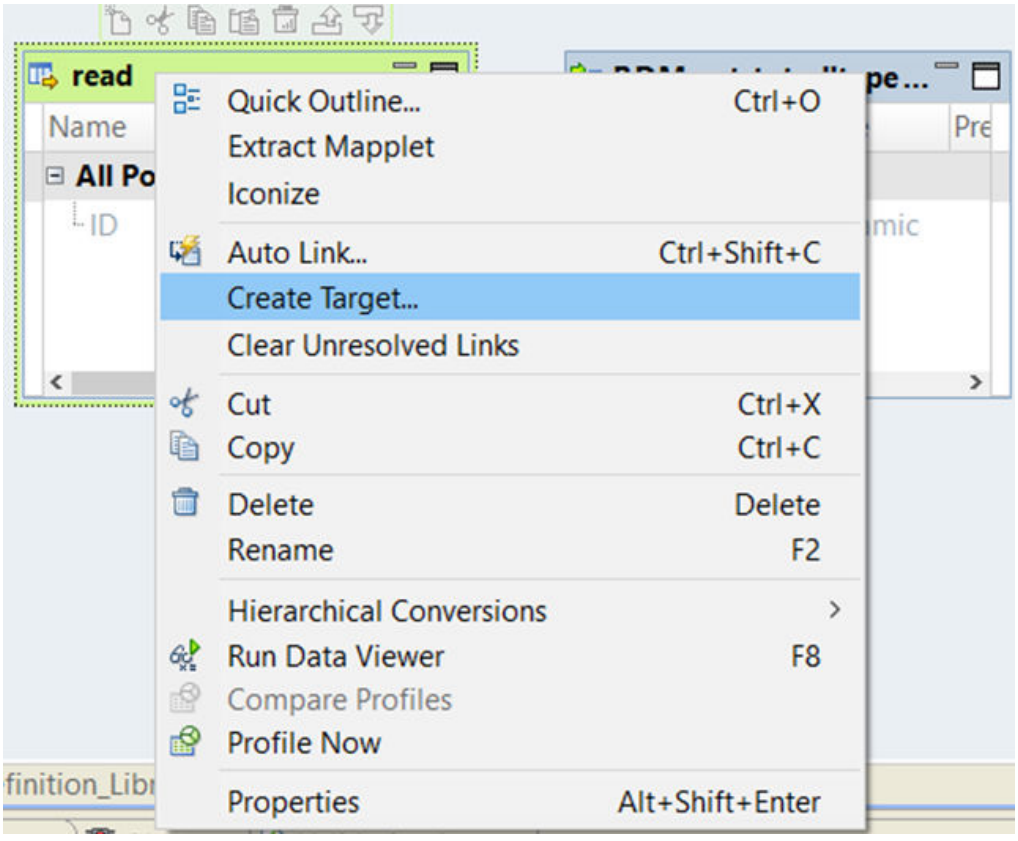
The Developer tool creates the data object operation for the selected data object.

Creating a PowerExchange for Kudu Target

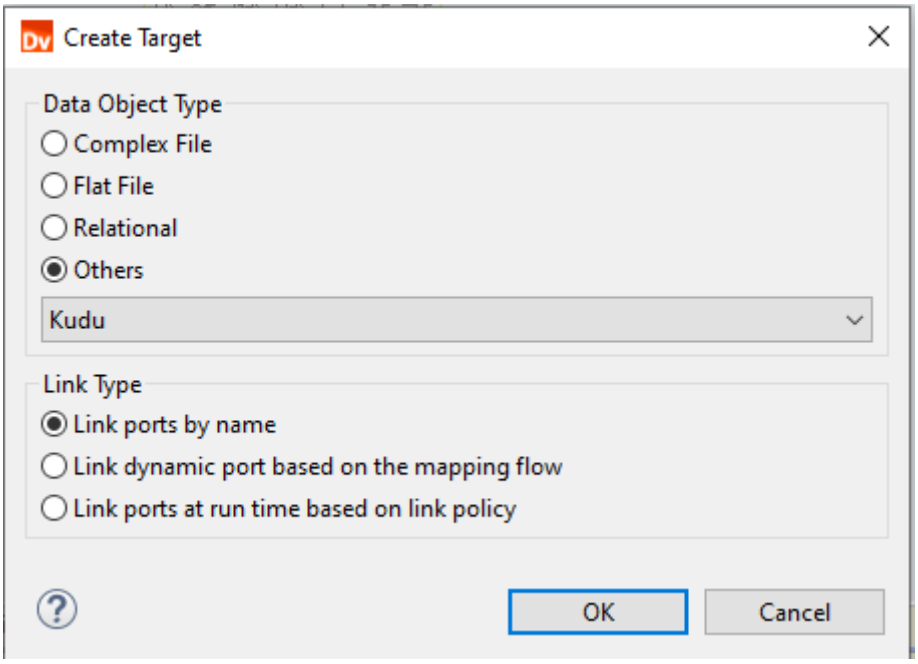
You can create a Kudu target using the **Create Target** option.

1. Select a project or folder in the **Object Explorer** view.
2. Select a source or a transformation in the mapping.
3. Right-click the Source transformation and select **Create Target**.
   The **Create Target** dialog box appears.

   The following image shows the **Create Target** option:

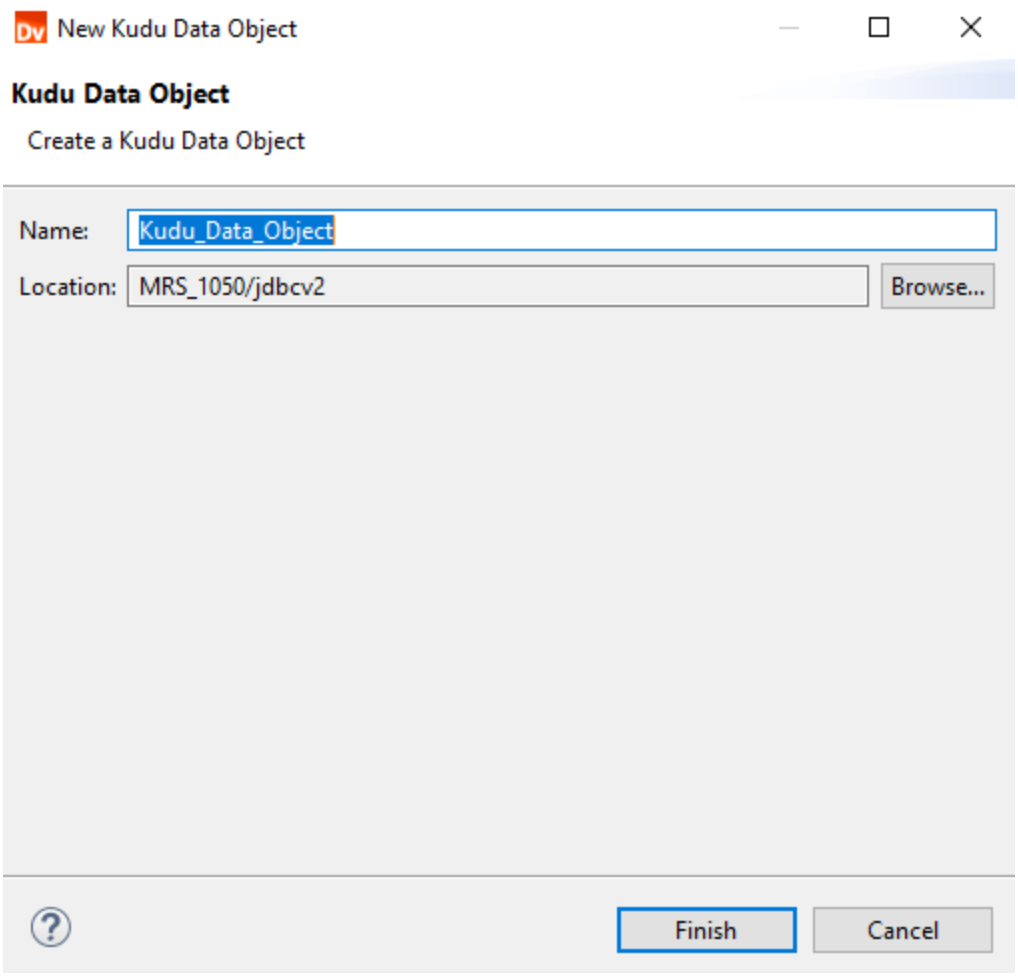


4. Select **Others** and then select **Kudu** data object from the list in the **Data Object Type** section.
   The following image shows the **Data Object Type** section:



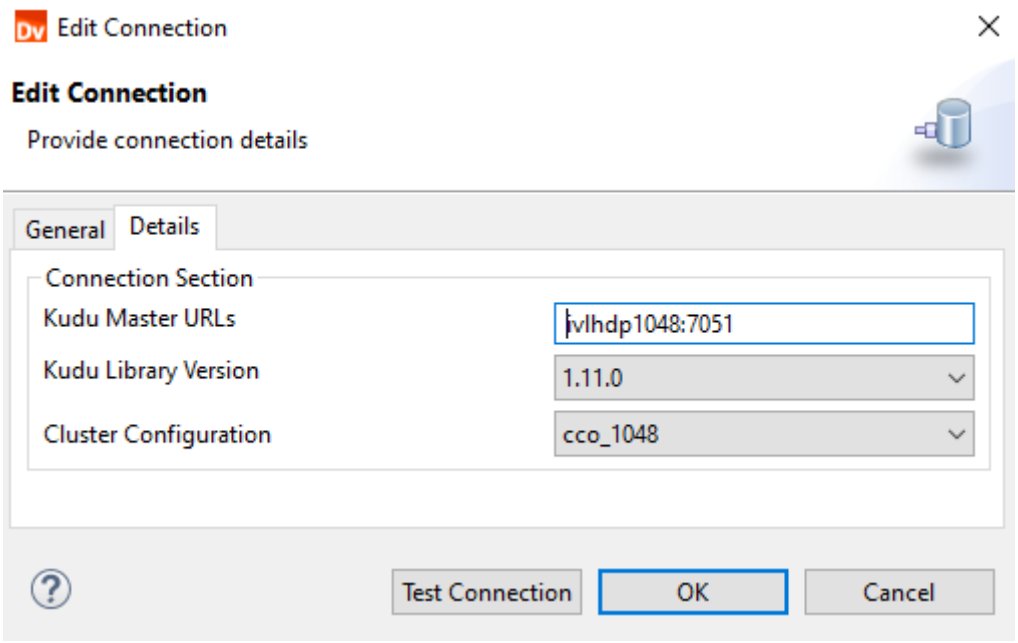
5. Click **OK**.
   The **New Kudu Data Object** dialog box appears.

   The following image shows the **New Kudu Data Object** dialog box:

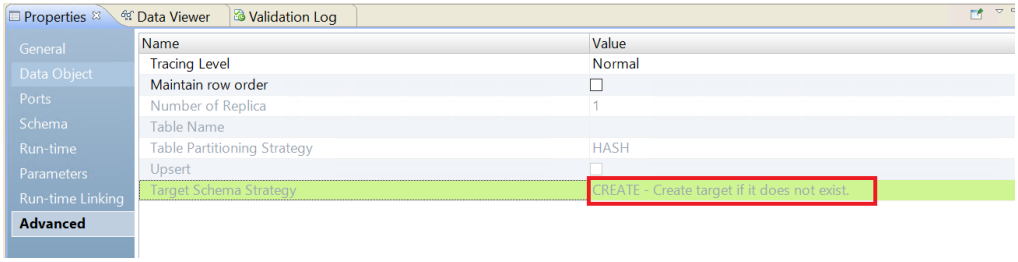


6. Enter a name for the data object.
7. Click **Finish**.
   The new target appears under the **Physical Data Objects** category in the project or folder in the **Object Explorer** view.
8. In the Kudu connection **Details** tab, specify the connection details to create the target type.

The following image shows the connection properties:



9. In the Kudu advanced target properties, select the target schema strategy.
The following image shows the configured the target schema strategy as **Create - Create target if it does not exist** in advanced properties:



Rules and Guidelines for PowerExchange for Kudu Data Objects

Use the following rules and guidelines when you create a mapping:

- You must define a primary key in the target table.
- You cannot parameterize the number of replicas for a Kudu target object.
- You cannot parameterize the connection when you use create target in a Kudu object.
- When you a use Kudu target object in a dynamic mapping, you cannot use an Update Strategy transformation to update, upsert, or delete the target table.

CHAPTER 5

PowerExchange for Kudu Mappings

This chapter includes the following topics:

PowerExchange for Kudu Mappings Overview

After you create a Kudu data object write operation, you can create a mapping to load data to a Kudu target.

You can define properties in an operation to determine how the Data Integration Service must load data to a Kudu target. You can load data to one or more Kudu targets. When the Data Integration Service loads data to the target, it converts the data based on the data types associated with the target.

Mapping Validation and Run-time Environments

You can validate and run mappings on the Spark engine in the Hadoop environment.

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.

Spark Engine

When you select the Hadoop environment, the Data Integration Service pushes the mapping to a compute cluster and processes the mapping on a Spark engine. The Data Integration Service generates an execution plan to run mappings on the Spark engine.

PowerExchange for Kudu Mapping Example

Your organization has a large amount of customer data from across regions stored in flat files. You organization needs to analyze data in the APAC region. Create a mapping that reads all the customer records from the flat file and write those records to a Kudu table.

You can use the following objects in a Kudu mapping:

Flat file input

The input file is a flat file that contains customer names and their details.

Create a flat file data object. Configure the flat file connection and specify the flat file that contains the customer data as a resource for the data object. Use the data object in a mapping as a read data object.

Kudu output

Create a Kudu data object write operation. Configure the Kudu connection and specify the Kudu object as a target for the data object. Use the data object in a mapping as a target data object.

When you run the mapping, the Data Integration Server reads customer records from the flat file and writes to the Kudu table.

Kudu Dynamic Mapping Overview

You can use Kudu data objects as dynamic targets in a mapping.

Use the Kudu dynamic mapping to accommodate changes to target and transformation logics at run time. You can use a Kudu dynamic mapping to manage frequent schema or metadata changes or to reuse the mapping logic for data sources with different schemas. Configure rules, parameters, and general transformation properties to create the dynamic mapping.

If the data source for a target changes, you can configure a mapping to dynamically get metadata changes at runtime. If a target changes, you can configure the Write transformation to accommodate changes to the target.

You do not need to manually synchronize the data object and update each transformation before you run the mapping again. The Data Integration Service dynamically determine transformation ports, transformation logic in the ports, and the port links within the mapping.

There are the two options available to enable a mapping to run dynamically. You can select one of the following options to enable the dynamic mapping:

- In the **Data Object** tab of the data object write operation, select the **At runtime, get data object columns from data source** option when you create a mapping.
  When you enable the dynamic mapping using this option, you can refresh the target schemas at the runtime.
- In the **Ports** tab of the data object write operation, select the value of the **Columns defined by** property as **Mapping Flow** when you configure the data object write operation properties.

For information about dynamic mappings, see the *Informatica Developer Mapping Guide*.

Refresh Schema

You can refresh the target schema at the runtime when you enable a mapping to run dynamically. You can refresh the imported metadata before you run the dynamic mapping.

You can enable a mapping to run dynamically using the **At runtime, get data object columns from data source** option in the **Data Object** tab of the Write transformations when you create a mapping.

When you add or override the metadata dynamically, you can include all the existing target objects in a single mapping and run the mapping. You do not have to change the source schema to update the data objects and mappings manually to incorporate all the new changes in the mapping.

You can use the mapping template rules to tune the behavior of the execution of such pipeline mapping.

When the Target transformation contains updated ports such as changes in the port names, data types, precision, or scale, the Data Integration Service fetches the updated ports and runs the mapping dynamically. You must ensure that at least one of the column name in the source or target file is the same as before refreshing the schema to run the dynamic mapping successfully.

Note: If refresh schema is enabled for a target transformation, you cannot perform upsert, update, and delete operations.

Even though the original order of the target ports in the table changes, the Data Integration Service displays the original order of the ports in the table when you refresh the schemas at runtime.

If there are more columns in the source file as compared to the target file, the Data Integration Service does not map the extra column to the target file and loads null data for all the unmapped columns in the target file.

If the Source transformation contains updated columns that do not match the Target transformation, the Data Integration Service does not link the new ports by default when you refresh the source or target schema. You must create a run-time link between the transformations to link ports at run time based on a parameter or link policy in the **Run-time Linking** tab and update the target schema manually. For information about run-time linking, see the *Informatica Developer Mapping Guide*.

Mapping Flow

You can add all the Source transformation or transformation ports to the target dynamically when you enable a mapping to run dynamically using the **Mapping Flow** option. You can then use the dynamic ports in the Write transformation.

When you select the **Mapping Flow** option, the Data Integration Service allows the Target transformation to override ports of the Write transformation with all the updated incoming ports from the pipeline mapping and loads the target file with the ports at runtime.

The Data Integration Service creates the target files dynamically based on the metadata of the incoming ports from the pipeline mapping.

To enable a dynamic mapping using the **Mapping Flow** option, select the value of the **Columns defined by** property as **Mapping Flow** in the **Ports** tab in the Write transformation.

Target Schema Strategy

You can choose to retain an existing target table or create a new target table in the target when you run a dynamic mapping.

You can select one of the following options in the **Target Schema Strategy** advanced properties for the data object write operation:

RETAIN - Retain existing target schema

The Data Integration Service retains the existing target schema.

Note: When you select **RETAIN** option and if the target table does not exist or the metadata of the source and target tables do not match, the mapping fails.

CREATE - Create table at run time

The Data Integration Service creates a new table based on the data object or the mapping flow if the table does not exist in the target.

When the Data Integration Service creates a table based on the data object, the table contains columns that match the ports in the data object. When the Data Integration Service creates a table based on the mapping flow, the table contains columns that match generated ports in the Write transformation.

Note: When you select the **CREATE** option and if the table already exists in the target location and the source object contains more columns than the existing target table, you must manually delete the existing table from Kudu. After you delete the existing table, you can run the mapping to create the target successfully.

Assign Parameter

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

CHAPTER 6

PowerExchange for Kudu Data Type Reference

This chapter includes the following topics:

Data Type Reference Overview

Developer Tool uses the following data types in Kudu mappings:

- Kudu native data types. Kudu data types appear in Kudu definitions in a mapping.
- 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. They appear in all transformations in a mapping.

PowerExchange for Kudu and Transformation Data Types

The following table lists the Kudu data types that the Developer Tool supports and the corresponding transformation data types:

Kudu Data Type	Transformation Data Type	Range and Description
Binary	Binary	Maximum value: 8,388,60 Default value is 8,388,60.
Boolean	String	A Boolean attribute.
String	String	1 to 104,857,600 characters. Default precision for String is 65536.

Kudu Data Type	Transformation Data Type	Range and Description
INT8	Int8	8-byte integer between -9,223,372,036,854,775,808 and +9,223,372,036,854,775,807. The length is set at 19 positions.
INT16	Integer	-2,147,483, 648 to 2,147,483,647, Precision 10, scale 0.
INT32	Integer	-2,147,483,648 to 2,147,483,647, Precision 10, scale 0.
INT64	Integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0.
Unixtime_Micros	Datetime	Date and time values.
Date	Datetime	Date and time values.
Float	Double	Floating point numbers with double-precision (64 bit). Maximum value: 1.7976931348623158e+307 Minimum value: -1.79769313486231E+307
Double	Double	
Decimal	Decimal	Decimal value with declared precision and scale. Scale must be less than or equal to precision.

INDEX